Diplomarbeit

Untersuchung der Bildkompression mittels der Lapped Hartley Transformation

von

Stephan May

Fachbereich Photoingenieurwesen und Medientechnik
Fachhochschule Köln

Referent: Prof. Dr. rer. nat. Dietmar Kunz

Korreferent: Prof. Dr.-Ing. Gregor Fischer

Fachhochschule Köln

Juli 2006

Thesis

Analysis of image compression using the Lapped Hartley Transform

by

Stephan May

Department of Imaging Sciences and Media Technology
University of Applied Sciences Cologne

First Reviewer: Prof. Dr. rer. nat. Dietmar Kunz

Second Reviewer: Prof. Dr.-Ing. Gregor Fischer

University of Applied Sciences Cologne

July 2006

Zusammenfassung

Titel: Untersuchung der Bildkompression mittels der Lapped Hartley Transformation

Autor: Stephan Ilja May

Referenten: Prof. Dr. rer. nat. D. Kunz / Prof. Dr.-Ing. G. Fischer

Die *Lapped Hartley Transformation* (LHT) ist eine neue reelwertige Transformation zweidimensionaler Bilddaten unter Verwendung überlappender Basisfunktionen. In dieser Arbeit wird ihr Einsatz als Alternative zur diskreten Kosinustransformation (DCT) für die Bildkompression nach dem JPEG-Verfahren untersucht. Die LHT wird hierzu an verschiedenen Bildbeispielen mit anderen Transformationen (DCT, MLT, LDT) quantitativ und visuell verglichen. Außerdem wird die Korreletion der Koeffizienten untersucht. Hierbei kommen die frei verfügbare Bildverarbeitungssoftware *ImageJ* und bereits vorhandene Java-basierte Implementierungen der untersuchten Kompressionsverfahren zum Einsatz.

Stichwörter: LHT, Bildkodierung, Kompression, Überlappende Transformation

Abgabedatum: 04. Juli 2006

Abstract

Title: Analysis of image compression using the Lapped Hartley Transform

Author: Stephan Ilja May

Reviewers: Prof. Dr. rer. nat. D. Kunz / Prof. Dr.-Ing. G. Fischer

The *Lapped Hartley Transform* (LHT) is a new real-valued transform of two-dimensional image data using lapped basis functions. In this thesis, its application as alternative to the discrete cosine transform (DCT) in image compression according to JPEG is investigated. To this end, the LHT is compared both quantitatively and visually with other transforms (DCT, MLT, LDT) on several sample images. Moreover, the correlation of the transform coefficients is determined. The study uses the publicly available image processing software *ImageJ* and already existing Java-based implementations of the applied compression methods.

Keywords: LHT, Image Coding, Compression, Lapped Transform

Date: 04. July 2006

Inhaltsverzeichnis

Glossar

Ausgewählte Fachausdrücke und Schlagwörter wurden in ihrem originalen Sprachgebrauch belassen, welche sich im Text durch eine kursive oder einer kapitalisierten Schrift hervorheben.

Abkürzungen

AC	Alternating Current
Bit	Binary Digit
DC	Direct Current
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DHT	Discrete Hartley Transform
DPCM	Differentiell Pulscode Modulation
ITU	International Telecommunication Union
JFIF	JPEG File Interchance Format
JPEG	Joint Photographic Expert Group
KF	Kompressionsfaktor
LDT	Lapped Directional Transform
LHT	Lapped Hartley Transform
MCU	Minimum Coded Unit
MLT	Modulated Lapped Transform
MSE	Mean Square Error
Pixel	Picture Element
PSNR	Peak-Signal-To-Noise-Ratio
RLC	Run Lenght Coding
TIF	Tagged Image File

1 Einleitung

Die folgende Arbeit beschäftigt sich mit der Bildkodierung von digitalen Einzelbildern unter Verwendung überlappender Transformationen.
Ein entscheidender Vorteil überlappender Basisfunktionen gegenüber einfachen Blocktransformationen, wie der *Diskreten Kosinustransformation* (DCT) des weit verbreiteten *JPEG File Interchange Format*s (JFIF) [15], ist die Vermeidung von Blockartefakten bei starker Komprimierung des Bildes. Des weiteren können derartige Transformationen auch effizient in anderen Bereichen der Signalverarbeitung eingesetzt werden, wie der Spektralanalyse, der Signalfilterung oder der Sprachkodierung [1].
Ein bereits genutztes Verfahren ist die *Modulated Lapped Transform* (MLT) [1], welche die Grundlage für die hier vorkommenden Verfahren bildet. Eine weitere überlappende Transformationsart stellt die *Lapped Directional Transform* (LDT) [8] dar, die in der Bildrestauration und in der medizinischen Bildgebung erfolgreiche Anwendung fand.
Das hier vorgestellte Verfahren, die *Lapped Hartley Transform* (LHT), ist eine neue Entwicklung von Professor Kunz. Ziel der Überlegungen galt eine Verkleinerung der Datenrate gegenüber der LDT zu realisieren, sowie die Eigenschaft der eindeutigen Orientierungsdetektion aus dem spektralen Leistungsspektrum zu ermöglichen.
Im Rahmen dieser Arbeit wird zunächst kurz auf elementare Grundlagen digitaler Bildverarbeitung eingegangen und ein Überblick der mathematischen Vorraussetzungen gegeben. Das nächste Kapitel erläutert den Quelltext des Programms hinsichtlich der Implementierung in Java und geht auf die neue Transformation näher ein. Anschließend wird diskutiert, wie die Verfahren arbeiten und wie die Kompressionsergebnisse bewertet werden können. Zum Schluß wird ein kurzer Ausblick in Hinblick auf weitere Untersuchungen gegeben.

2 Grundlagen

2.1 Bildkodierung

Digitale Signale lassen sich mit Hilfe von mathematischen Rechenoperationen, den sogenannten Algorithmen, einfach be- und verarbeiten. Dazu gehört auch die Reduktion und Kompression von Daten. Ein Computerbild liegt immer als eine Menge von ganzzahligen, zeit- und wertdiskreten Binärwörten vor, so daß es durch eine zweidimensionale Funktion $G(u, v)$ beschrieben werden kann. Um einen möglichst prägnanten Datenstrom zu erhalten, der bei der Speicherung nicht unnötig viel Kapazität in Anspruch nimmt, ist eine anwendungsspezifische Komprimierung vonnöten.

2.1.1 Verlustbehaftete und verlustfreie Kompression

Der Informationsgehalt eines Signals wird bemessen nach Häufigkeit und Auftreten bestimmter Ereignisse, d.h. selten auftretende Werte müssen mit mehr Bits kodiert werden als Häufigere. Dazu beseitigt man irrelevante Daten, die vom Empfänger nicht wahrgenommen werden können, und Redundante, welche durch geschicktes Verarbeiten keine neuen Informationen bringen würden. Eine vollständige Rekonstruktion des Originalbildes ist mittels einer verlustbehafteten (engl. *lossy*) Kompression nicht möglich, bei verlustloser (engl. *lossless*) hingegen schon. Eine höhere Speicherplatzfreigabe wird bei verlustbehafteten Verfahren erziehlt. Darüberhinaus können Bilder mit einer kleinen Dateigröße schneller geladen oder mittels eines Senders und Empfängers schneller übertragen und empfangen werden, z.B. über das Internet.
Im folgenden soll kurz auf einige Eigenschaften eingegangen werden, welche allen Kompressionsprozessen in der Bildverarbeitung gleich sind.

2.1.2 Unterabtastung und Quantisierung

Eine erste Datenreduktion läßt sich durch Unterabtastung der Chrominanzwerte erziehlen, da das menschliche Auge für Helligkeitsunterschiede empfindlicher reagiert gegenüber Farbschwankungen [7]. Je nachdem welches Abtastverhältnis[1] vorliegt, wird nur jeder m-te Farbwert übernommen. Die restlichen Werte gehen verloren. Durch die spätere Quantisierung der transformierten Pixelwerte, wird jeder Wert nach entsprechenden Quantisierungsmatrizen[2] auf eine möglichst geringe Anzahl von Bits abgebildet, wobei hohe Ortsfrequenzen mit weniger Bits gespeichert werden als Tiefe, auf Grund des begrenzten Auflösungsvermögens des Auges. Dazu gibt es unterschiedliche Methoden, wie skalare Quantisierung, gleichmäßige oder sukzessive Approximation oder wahrnehmungsoptimierte Vektorquantisierung [4].

2.1.3 Entropiekodierung und Lauflängenkodierung

Die eigentliche Kodierung des Signals ist dagegen wiederum verlustfrei. In der *Präkodierung* werden die Symbole als abhängig voneinander betrachtet, mit dem Ziel einer Verminderung der Intersymbolredundanz, d.h. benachbarte Symbole werden durch Sortieren, Substitution oder Ähnlichkeiten miteinander korreliert, um Bits einzusparen.
In den hier verwendeten Verfahren werden die wechselanteiligen *Alternating Current* (AC) Koeffizienten mit einer *Lauflängenkodierung* (RLC) verarbeitet. Identische Werte werden zu einer Kette zusammengefasst, d.h. man benötigt den Anfang und die Länge dieser Folge von Zahlen. Die Reihenfolge der AC-Werte geschieht nach einer *Zick-Zack* Umsortierung, denn so stehen möglichst viele gleiche Werte hintereinander, die beim Quantisieren zu null wurden. Die *Direct Current* (DC) Werte werden seperat mit einem *Differenzverfahren* (DPCM) kodiert, bei der nur der Wert der Differenz zu seinem Vorgänger übertragen wird.
Abschließend werden in der *Entropiekodierung* die Symbole als unabhängig voneinander betrachtet und in neue Codewörter geschrieben, welchen wiederum, je nach Häufigkeit oder Wahrscheinlichkeit, nur so viele Bits wie erforderlich zugeordnet werden. Dies kann mit vordefinierten Tabellen gelöst werden, die im

[1] siehe Anhang A1
[2] Die verwendeten Tabellen im Quelltext entsprechen denen aus der ITU-T81 Norm.

Datenstrom mitgeliefert werden müssen. Als Repräsentant dieser Gruppe steht der *Huffman-Coder*.

2.2 Spektrale Transformation

Ein fundamentaler Formalismus ist maßgebend für die Verarbeitung von digitalen Bilddaten: die harmonische Analyse von Signalen. Jede Funktion kann durch eine Summe von Sinus- und Kosinusfunktionen beschrieben werden. Der wohl wichtigste Grund für die Beschäftigung mit der *Fouriertransformation* liegt darin, daß sie geeignet ist, lineare, zeitinvariante Systeme zu beschreiben [17]. Gemeint ist damit, daß auf eine Veränderung eines ursprünglichen Eingangssignals, das andere Ausgangssignal entsprechend der Veränderung reagiert. Mit Hilfe der Spektraltechnik werden gleiche oder ähnliche Strukturen in einem Bild in nur wenigen Koeffizienten ausgedrückt, weshalb sie für die Komprimierung hervorragend geeignet ist. Das Fourierspektrum gibt demnach an, welche Frequenzen zu welchen Anteilen im Signal vorhanden sind. Umgekehrt kann die ursprüngliche Funktion aus dem Fourierspektrum durch inverse Rücktransformation wieder eindeutig rekonstruiert werden. Die Überführung der Daten eines Bildes ins Spektrum und die anschließende Rücktransformation in den Originalbereich ist abgesehen von Rundungsfehlern verlustfrei.
Die Vorwärtstransformation für ein diskretes, periodisches, zweidimensionales Signal ist definiert als

$$G(m,n) = \frac{1}{\sqrt{MN}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} g(u,v) \cdot e^{-i2\pi(\frac{mu}{M}+\frac{nv}{N})} \ , \tag{2.1}$$

wobei der komplexwertige Exponentialtherm umgeschrieben werden kann zu

$$e^{-i2\pi(\frac{mu}{M}+\frac{nv}{N})} = \cos\left[2\pi\left(\frac{mu}{M}+\frac{nv}{N}\right)\right] + i\sin\left[2\pi\left(\frac{mu}{M}+\frac{nv}{N}\right)\right]. \tag{2.2}$$

Zur Berechnung der *Diskreten Fouriertransformation* (DFT) existieren schnelle Algorithmen, in denen die Abfolge der Berechnungen so ausgelegt ist, daß gleich-

artige Zwischenergebnisse nur einmal berechnet und in optimaler Weise mehrfach wiederverwendet werden [2].
Für die Implementierung zweidimensionalener Transformationen macht man sich die Eigenschaft zu Nutze, den Transformationskern in ein Produkt von zwei eindimensionalen Funktionen zu zerlegen. In der Regel realisiert man das durch einen *Zeilen/Spalten-Algorithmus*. Diese Idee vereinfacht das Programm, da man nur einen Algorithmus braucht, der zuerst zeilenweise und danach spaltenweise auf die Zeilenkoeffizienten angewandt wird.
Jede Spektraltransformation arbeitet mit vordefinierten Basisfunktionen, die das Bildsignal nach tiefen und hohen Frequenzen ordnen und ausdrücken. Aus N^2 Werten werden N^2 Basissignale, die jeweils eine der N^2 Frequenzen enthalten [9]. Damit es an den Blockgrenzen[3] nicht zu *Leakage*-Defekten kommt, kann der Wellenanteil mit einer geeigneten *Fensterfunktion* (engl. *Windowing*) mulipliziert werden. Diese „hüllt“ alle Funktionen ein und läßt sie zum Rand hin weich abklingen.

2.2.1 Die Basisfunktionen von DCT, MLT und LDT

Die *Diskrete Kosinustransformation* ist eine reelwertige Methode, die Bilddaten in ihr Spektrum überzuführen. Das Signal wird hier nicht mit einer Fensterfunktion multipliziert, sondern zu den Rändern hin gespiegelt, so daß sich das Signal in umgekehrter Reihenfolge wiederholt.
Die zweidimensionale DCT hat folgende mathematische Form:

$$G(m,n) = \frac{2}{\sqrt{MN}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} g(u,v) \cdot c_m \cos\left(\frac{\pi(2u+1)m}{2M}\right) \cdot c_n \cos\left(\frac{\pi(2v+1)n}{2N}\right), \quad (2.3)$$

für $0 \leq m,n < M,N$
$c_m, c_n = \frac{1}{\sqrt{2}}$, für $m,n = 0$
$c_m, c_n = 1$, sonst.

M und N sind beim JPEG Verfahren gleich acht, d.h. der zu transformierende Block besteht aus 64 Werten.

[3] In der Regel besteht ein Block aus acht mal acht Pixeln.

Die *Modulated Lapped Transform* [1] zerlegt das Bildsignal in eine Blockgröße von 2L Bildpunkten, wobei der Überlappungsbereich aneinander liegender Blöcke in jeder Richtung die Hälfte der Blockgröße ist. Jeder Pixel wird insgesamt viermal von einem Block erfasst. Damit die tiefen Frequenzen nicht in die Hohen hineinlecken, werden die Wellenfunktionen der MLT mit einer sinusförmigen Fensterfunktion multipliziert, die eine Periode von 4L hat. Das Ergebnis der Transformation sind 64 Koeffizienten, die sich aus der Summe der Produkte von Bildsignal und Basisfunktionen ergeben.
Die Formel der zweidimensionalen MLT Basisfunktionen lautet:

$$\begin{aligned} p(k,l,n,m) &= w(k,n) \cdot w(l,m) = \\ &= h(n) \cdot c(k,n) \cdot h(m) \cdot c(l,m) = \\ &= \sqrt{\frac{2}{L}} \sin\left[\frac{\pi}{2L}\left(n+\frac{1}{2}\right)\right] \cdot \cos\left[\frac{\pi}{L}\left(n-\frac{L-1}{2}\right)\left(k+\frac{1}{2}\right)\right] \cdot \\ &\quad \sqrt{\frac{2}{L}} \sin\left[\frac{\pi}{2L}\left(m+\frac{1}{2}\right)\right] \cdot \cos\left[\frac{\pi}{L}\left(m-\frac{L-1}{2}\right)\left(l+\frac{1}{2}\right)\right] \end{aligned} \tag{2.4}$$

für $k,l = 0,...,L-1; n,m = 0,...,2L-1$

Die Indizes k und l geben die spektrale Frequenz an, m und n die Kooradinaten im Spektrum, L die Blockgröße.
Die *Lapped Directional Transform* [8] setzt sich aus einer weiteren MLT zusammen, die entgegengesetzte Symmetrieeigenschaften hat und mit MLT′ bezeichnet wird.

$$w'(k,n) = h(n) \cdot \sin\left[\frac{\pi}{L}\left(n-\frac{L-1}{2}\right)\left(k+\frac{1}{2}\right)\right] \tag{2.5}$$

Dies führt zu einer doppelten Anzahl zu speichernder Koeffizienten, jedoch wird eine eindeutige Bestimmung von Orientierungen im Spektrum ermöglicht durch Erzeugen eines neuen Spektrums, welches sich aus einer Summen- und Differenzbildung gemäß Abbildung 2.1 ergibt. Es ensteht eine Fallunterschiedung für negative und positive Frequenzen.
Formal besteht das Spektrum aus $4L^2$ Koeffizienten, welche sich auf $2L^2$ unabhängige Werte reduzieren lassen. Um das Signal eindeutig zu rekonstruieren, würde

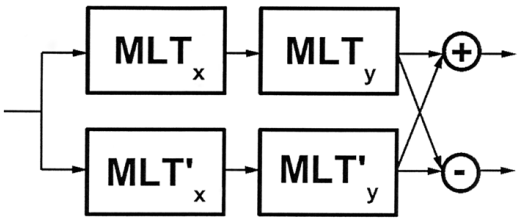


Abb. 2.1: *Blockdiagramm der LDT*

es genügen, mit der MLT oder mit der MLT′ zu arbeiten, da sie zueinander gespiegelt sind und sich nur auf Grund des Vorzeichens für ungerade Frequenzen unterscheiden.

2.2.2 Die Basisfunktionen der LHT

Neben der komplexwertigen Fourieranalyse gibt es noch andere Techniken, ein Signal vom Ortsbereich in den Frequenzbereich zu transformieren: Die *Hartley Tansformation* ist daher für die Anwendung in der digitalen Bildverarbeitung von zunehmender Bedeutung, da sie eine einfache, reellwertige Methode ist mit der Eigenschaft, daß Vorwärts- und Rückwärtstransformation (bis auf einen Faktor) identisch sind. Das Programm *ImageJ*[4] verwendet für die Spektrumsberechnung den *Hartley Transformationsformalismus*, der vereinfacht geschrieben eine Addition aus Kosinus mit Sinus darstellt.

$$\mathrm{cas}(\alpha) = \cos(\alpha) + \sin(\alpha) \tag{2.6}$$

Die Hartley Transformation ist eine orthogonale Transformation, d.h. die Basisvektoren stehen paarweise senkrecht aufeinander und sind linear unabhängig, so daß die transponierte Matrix A^T gleich ihrer Inversen A^{-1} ist.
Die zweidimensionale *Diskrete Hartley Transformierte* (DHT) wird beschrieben mit

$$H(m,n) = \frac{1}{\sqrt{MN}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} h(u,v)\,\mathrm{cas}\left(\frac{2\pi mu}{M} + \frac{2\pi nv}{N}\right). \tag{2.7}$$

[4] http://rsb.info.nih.gov/ij/

Aus den DHT Koeffizienten lassen sich die DFT Koeffizienten jederzeit berechnen [5]. Ein charakteristischer Vorteil ist die Symmetrieeigenschaft der DHT Matrix, so daß für den Enkoder als auch den Dekoder dieselbe Rechenroutine verwendet werden kann [6].
Im Vergleich zu den Basisfunktionen der DCT oder MLT ist die LHT (wie auch die LDT) nicht separierbar, d.h. die Berechnung des zweidimensionalen Bildsignals kann nicht allein durch eine Multiplikation zweier eindimensionaler Signale berechnet werden. Da sich aber die LHT Koeffizienten aus der separierbaren MLT herleiten lassen, ist das Ergebnis eine Summe von vier Teiltransformationen, welche sich aus dem trigonometrischen Additionstheorem

$$\begin{aligned} 2\cos(\alpha+\beta) = &\cos(\alpha)\cos(\beta) \\ &+ \cos(\alpha)\cos(-\beta) \\ &+ \cos(-\alpha)\cos(\beta) \\ &- \cos(-\alpha)\cos(-\beta) \end{aligned} \tag{2.8}$$

ergeben.
Um auf eine Form der zweidimensionalen LHT gemäß Gleichung 2.7 zu kommen, wird mittels des Zeilen/Spalten-Algorithmusses zweimal eine eindimensionale Transformation durchgeführt, einmal in horizontaler und einmal in vertikaler Richtung. Mit Hilfe von Gleichung 2.8 kann das Spektrum aus vier korrespondierenden Werten gebildet werden, die symmetrisch zum Ursprung im gleichen Abstand angeordnet sind.
Die allgemeine Formel der zweidimensionalen LHT Basisfunktionen für alle Werte lautet:

$$F(x,y) = \cos(\pi x)\cdot\cos(\pi y)\cdot\mathrm{cas}(2\pi(f_x\cdot x + f_y\cdot y))\ , \tag{2.9}$$

wobei $\cos(\pi x)$ und $\cos(\pi y)$ die Fensterfunktionen in horizontaler und vertikaler Richtung dastellen, mit denen die Wellenfunktionen multipliziert werden, um den Überlappungsbereich weich abklingen zu lassen.
In der Schreibweise mit Abhängigkeit von den Ortsfrequenzen k, l und den Orts-

koordinaten n, m, formuliert sich Gleichung 2.9 um zu

$$\begin{aligned} H_p(k,l,n,m) &= h(n) \cdot h(m) \cdot \mathrm{cas}(k,l,n,m) = \\ &= \sqrt{\frac{2}{L}} \sin\left[\frac{\pi}{2L}\left(n+\frac{1}{2}\right)\right] \cdot \sqrt{\frac{2}{L}} \sin\left[\frac{\pi}{2L}\left(m+\frac{1}{2}\right)\right] \cdot \\ &\mathrm{cas}\left[\frac{\pi}{L}\left(\left(n-L+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)+\left(m-L+\frac{1}{2}\right)\left(l+\frac{1}{2}\right)\right)\right] \end{aligned} \tag{2.10}$$

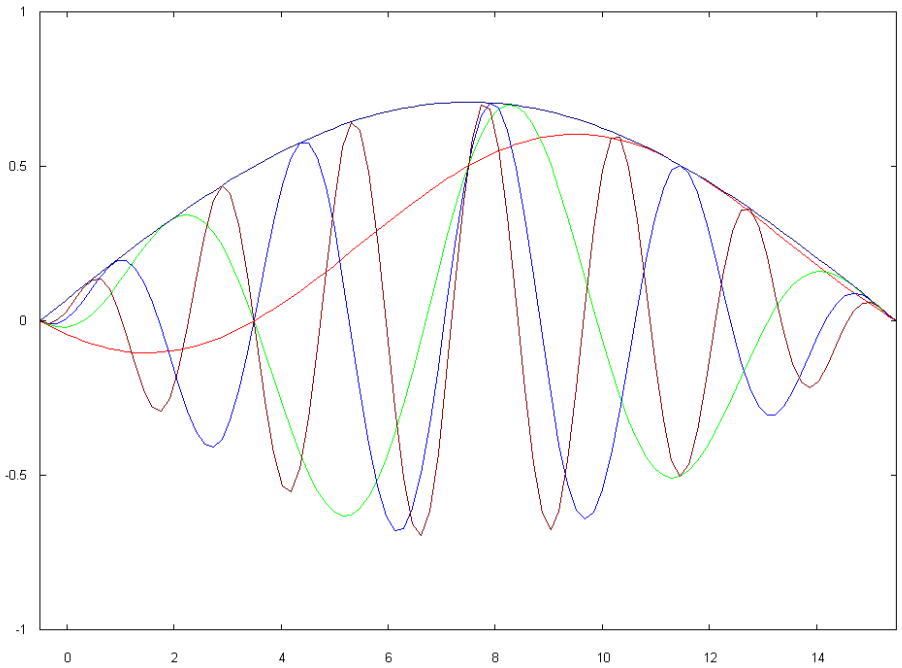


Abb. 2.2: *Die von h(n) modulierten eindimensionalen Basisfunktionen der LHT für gerade, positive Frequenzen k=0, k=2, k=4, k=6*

Die Funktionen haben für gerade, positive Frequenzen k einen Nulldurchgang bei $\frac{L-1}{2}$ (für L=8 bei 3,5), einen gemeinsamen Schnittpunkt bei $\frac{2L-1}{2}$ (für L=8 bei 7,5) und jeweils ein Maximum in $\frac{3L-1}{2}$ (für L=8 bei 11,5). Genauso könnte man diese MLT′ Funktionen aus allen ungeraden, negativen Frequenzen berechnen; sie sind dann bis auf das Vorzeichen identisch.
Für die geraden, negativen Frequenzen liegt der Nulldurchgang, entsprechend den MLT Funktionen, bei $\frac{3L-1}{2}$ (für L=8 bei 11,5) und die Maxima bei $\frac{L-1}{2}$ (für L=8 bei 3,5). Der gemeinsame Schnittpunkt bleibt bei $\frac{2L-1}{2}$ (für L=8 bei 7,5) bestehen.

Gleiches gilt wieder für die ungeraden, positiven Frequenzen mit umgekehrtem Vorzeichen.

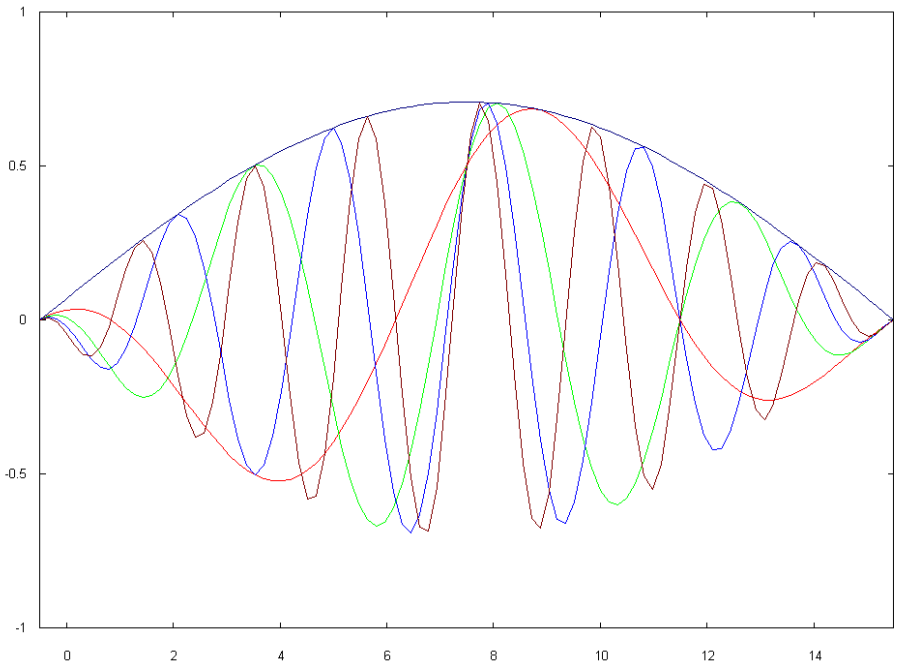


Abb. 2.3: *Die von h(n) modulierten eindimensionalen Basisfunktionen der LHT für ungerade, positive Frequenzen k=1, k=3, k=5, k=7*

Zwischen den Frequenzen k herrscht, wie aus obiger Formel 2.10 ersichtlich, ein Versatz von $\frac{1}{2}$.
Bei der Realisierung im Programmcode wird die zweidimensionale LHT aus den schon bekannten MLT und MLT′ Funktionen gebildet.

$$H(x,y) = MLT_x MLT_y + MLT_x MLT'_y + MLT'_x MLTy - MLT'_x MLT'_y \quad (2.11)$$

Oder anders formuliert zu:

$$\begin{aligned} H_p(k,l,m,n) &= w(k,n) \cdot w(l,m) \\ &+ w(k,n) \cdot w'(l,m) \\ &+ w'(k,n) \cdot w(l,m) \\ &- w'(k,n) \cdot w'(l,m) \end{aligned} \quad (2.12)$$

Durch die Verknüpfung von MLT und MLT′ Koeffizienten haben die Basisfunk-

tionen der LHT von Block zu Block entgegengesetzte Vorzeichen. Eine Fallunterscheidung, im Sinne der LDT, muß nicht vorgenommen werden, jedoch muß das Vorzeichen für negative Frequenzen berücksichtigt werden, so daß die negativen MLT′ Werte zu positiven werden.

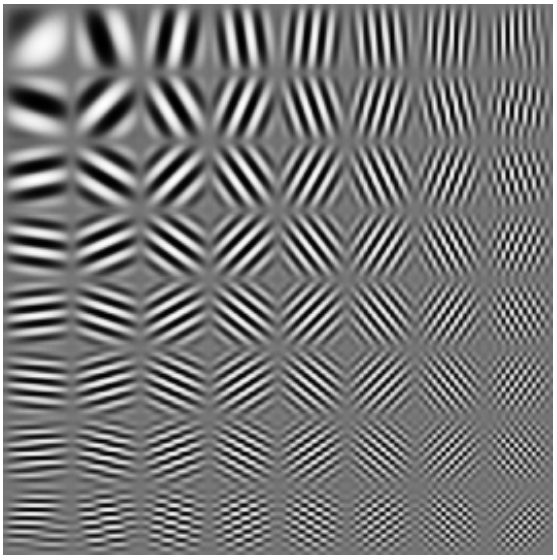


Abb. 2.4: *Die Basisfunktionen der LHT*

An dieser Stelle sei darauf hingewiesen, daß die LHT nicht vollständig orthogonal ist, da Hin- und Rücktransformation nicht das gleiche Bild wiedergeben. Das liegt daran, daß sie bezüglich der horizontalen und vertikalen Richtung orthogonal ist, nicht aber in ihren Diagonalen, da hier eine Phasenverschiebung von π aufritt, statt regulär von $\frac{\pi}{2}$. Dies führt bei der Rücktransformation zu Geisterbildern, die durch eine entsprechende Gegenfilterung seitens des Enkoders und Dekoders zu 100% weggefiltert werden können.

3 Der Programmcode

3.1 Die Java Klassen

Die komplette Realisierung des Programms[1] gliedert sich in mehrere Javadokumente. Zum einen in die erforderlichen ImageJ-PlugIn-Schnittstellen und zum anderen in die kodierungs- und transformations relevanten Quelltexte. Diese sind als ein eigenes Java Paket *package LappedTransform* zusammengefasst und werden im Ordner *LappedTransform* gefunden, welcher sich im Unterordener *plugins* vom Programmordner *ImageJ* befindet. Die Programmsegmente für die LHT wurden in die bestehenden Sourcecodes eingefügt.
Zum En- und Dekodieren von Bildern mit überlappenden Transformationen werden in ImageJ die PlugIns *JPG_Writer_.java* und *JPG_Reader_.java* aufgerufen. Durch den *JPG_Writer_* können die Einstellungen für die Wahl der Transformationsart, des Qualitätsfaktors und des Abtastverhältnisses in einem Auswahlfenster vorgenommen werden. Darüber hinaus gibt es die Möglichkeit, sich die Y, Cb, Cr Werte der LHT je in einem separaten Bild ausgeben zu lassen[2]. Das so erhaltene Spektrumsbild ist für die Korrelation von den Pixelwerten untereinander von Bedeutung. Mit dem *JPG_Reader_* lassen sich die kodierten Bilder wieder in ImageJ darstellen.
Damit die Wahl der entsprechenden Transformationsart gewährleistet ist, werden in *TransformationMethod.java* die vier unterschiedlichen Typen zu neuen Objekten erzeugt. Alle Transformationen greifen auf dieselben schnellen Algorithmen zurück, die in der *Jmlt.java* stehen, wie z.B. der Type-IV der eindimensionalen diskreten Sinus- und Kosinustransformation. Außerdem sind hier die Bausteine für die Berechnung der MLT und MLT′ festgelegt. Die Ränder müssen gesondert verarbeitet werden, da bei den überlappenden Transformationen die Blockgröße

[1] basiert auf der Diplomarbeitet von Sylvia Schmitt [10]
[2] Die Implementierung von *makeSpec* wurde von Professor Kunz vorgenommen.

2L mal 2L ist, also doppelt so groß wie bei der DCT. Es ist deshalb nötig, das Bild in drei Bereiche einzuteilen, so daß zwischen linkem Rand (MLT_LEFT), rechtem Bildrand (MLT_RIGHT) und mittlerem Bereich (MLT_WORK) eindeutig unterschieden werden kann. Die Berechnungsvorschriften dafür stehen in der *Jmlt.java*, der Methodenaufruf ist seperat aber in *MLTmethod.java* deklariert. Die *Jmlt* enthält auch die inverse Transformationsberechnung für die überlappenden Transformationen, da hier nur reellwertige Funktionen benötigt werden und deshalb auf dieselben schnellen Algorithmen zugegriffen werden kann.
In *JpegEncoder.java* werden die unterschiedlichen Berechnungsarten aufgerufen. Es ist gewissermaßen das Hauptprogramm, welches die Kodierung verwaltet und ermöglicht. Der *JpegDecoder.java* enthält entsprechend die Information zur Dekodierung.

3.1.1 Die LHT im Enkoder

Der Enkoder ist definiert in sechs lokale Klassen: der äußeren Hauptklasse *class JpegEncoder*, innerhalb derer fünf weitere innere Klassen, *class MLT*, *class DCT*, *class HuffmanItem*, *class Huffman* und *class JpegInfo* stehen. Damit hat man einen höchst flexiblen Mechanismus, so daß die inneren Klassen auf die Membervariablen der äußerern Klasse zugreifen können und umgekehrt [12].
JpegInfo beinhaltet die blockweise Einlesevorschrift der Originaldaten und die anschließende Konvertierung vom RGB Farbraum in den YCbCr Raum. Somit liegen die Daten getrennt voneinander in den drei Komponenten für Luminanz Y, und den Farbdifferenzsignalen Cb und Cr vor. Die sukzessive Verarbeitung der einzelnen Werte zu Dateneinheiten (*Data Units*), wird immer verschachtelt (engl. *interleaved*) angeordnet, d.h. eine *Minimun Coded Unit* (MCU) ist eine Folge der acht mal acht Bildpunkten großen Data Units der einzelnen Komponenten. Je nachdem welches Abtastverhältnis [3] vorliegt, besteht eine MCU aus mehr als nur einer Y Data Unit. Die spätere Entropiekodierung erfolgt jeweils für eine komplette MCU, bevor der fertige Bitsrom als *outStream* gespeichert werden kann. Die Quantisierung der spektralen Koeffizienten und die komplette Kodierung, ist für alle Verfahren gleich unter Zuhilfenahme von Tabellen. Dazu wurde die Klasse *class Huffman* von Professor Kunz umgeschrieben und mit der Klasse *class Huff-*

[3] siehe Anhang A1

manItem erweitert, da hier Fehlkodierungen auftraten auf Grund des zu kleinen Wertebereichs der Huffman Kategorienummer, d.h. wie viele Bits zum Übertragen der AC-Koeffizienten bzw. des Differenzbetrages des DC-Wertes benötigt werden.
In der Klasse *class DCT* befindet sich die Berechnung der DCT und die entsprechende Quantisierung, nach den dafür vorgesehenen Quantisierungsvorschriften für die Luminanz- und Chrominanzwerte nach Norm ITU-T81.
Die Klasse *class JpegEncoder* greift auf zwei wichtge Basisklassen des vordefinierten Pakets *java.io* zurück - dem schreibenden *OutputStream* und dem lesenden *InputStream*. BufferedOutputStream(out) puffert die Ausgabe in outStream. Die Methode *outStream.flush* sorgt dafür, daß die Daten dann tatsächlich geschrieben werden. Für den fertigen Bitstrom sind die *write*-Aufrufe von besonderer Bedeutung, da hier die Marker gesetzt werden, die dem Dekoder später mitteilen, wann das Bild beginnt und wann es endet. Dazu werden vor dem Bitstrom des Bildes ein *SOI-Marker* (Start of Image)[4] eingefügt und entsprechend am Ende der *EOI-Marker* (End of Image)[5]. Anzumerken ist, daß der komplette Bitstrom mit Markern versehen ist, da zum einen die verwendeten Tabellen mitgeliefert werden müssen und zum weiteren die einzelnen Scans mit Headern versehen sind, da der Jpeg Dekoder [15] standardisiert ist und somit wieder eine richtige Darstellung des Bildes möglich ist. Durch die Methode *Compress ()* wird die Enkodierung des Bildes eingeleitet, in der eine Abfrage für die entsprechende Transformationsart vorgenommen wird, damit auf die richtige Methode für den Einlesevorgang der DCT, sowie der MLT, LDT und LHT zugegriffen werden kann.
Für die Implementierung der LHT wurden in erster Linie die Klassen *class JpegEncoder* und *class MLT* verändert, da die drei Transformationsarten MLT, LDT und LHT soweit wie möglich den Code gemeinsam verwenden, so daß nur wenige Erweiterungen für das Funktionieren der LHT vorgenommen wurden.
Zunächst wird ein Überblick über den formalen Ablauf des Einlesevorgangs der überlappenden Transformationen gegeben, um dann im Detail auf den Zweck wichtiger Bausteine einzugehen.
Um eine scharfkantige Blockstruktur, wie im JPEG Verfahren, zu vermeiden, ist die Überlappung bestimmt durch die Verdopplung der Blockgröße und der Multiplikation der Daten, mit der zum Rand hin abfallenden Fensterfunktion, so daß

[4] in hexadezimaler Schreibweise FFD8
[5] in hexadezimaler Schreibweise FFD9

jeder Pixel Teil von vier Blöcken wird [8]. Damit dies an den Bildrändern ebenfalls gewährleistet ist, muß als Erstes der Randbereich vorbereitet werden. Dazu werden die ersten vier Zeilen $li \leq DCTSIZE/2$ des oberen Bildbereiches vollständig eingelesen und horizontal transformiert. Dabei ist für die Berechnung des linken und rechten Randes die einzulesende Spaltengröße ebenfalls $co \leq DCTSIZE/2$, im mittleren Bereich aber *DCTSIZE*. Dabei wird genau soviel Speicherplatz zur Verfügung gestellt, wie das nötige Abtastverhältnis es erfordert. Nach der Transformation in horizontaler Richtung, werden dann die Daten vertikal transformiert.

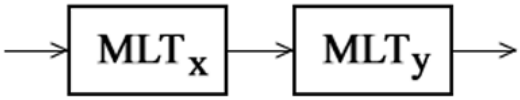


Abb. 3.1: *MLT Blockdiagramm*

Auch hier wird zwischen den drei Bildbereichen unterschieden. Da in Java ein zweidimensionales Array aus zwei eindimensionalen Arrays realisiert ist, wird die ursprüngliche zeilenweise Anordnung zu einer spaltenweisen Reihenfolge vertauscht, d.h. nun beginnt die äußere Laufschleife bei *co* statt bei *li*.
Damit ist die Vorbereitung des oberen Bildbereichs abgeschlossen, es liegen aber noch keine fertigen Ausgabedaten vor. Der überlappende Algorithmus ist so konzipiert, daß immer acht mal acht fertige Koeffizienten ausgegeben werden, ein komplett transformierter Block aber aus einer 16 mal 16 Pixelumgebung besteht. Dazu stehen immer vier Werte des letzten Blocks im Speicher und aus den acht neuen Koeffizienten des nächsten einzulesenden Blocks, der um *DCTSIZE/2* versetzt beginnt, werden davon die Hälfte der Werte die Neuen Alten. Am linken Rand bestehen die ersten acht resultierenden MLT Koeffizienten aus insgesamt zwölf Werten[6] und am rechten Rand werden aus den letzten vier Einzulesenden, zusammen mit den letzten acht des vorangegangenen Blocks, acht Koeffizienten ausgegeben. Eine weitere Unterscheidung muß für die letzten Zeilen bzw. Spalten vorgenommern werden, denn hier beträgt die Blockgröße wieder *DCTSIZE/2*.
An dieser Stelle sei nochmals darauf hingewiesen, daß nur Bilder mit einer durch acht teilbaren Bildhöhe und Bildbreite verarbeitet werden können.

[6] aus den vier Vorbereiteten, und den nächsten acht

Um die Bildverarbeitung auf diese Weise zu realiseren, benötigt man einige wichtige Schiebe- und Zwischenspeichermodule. Die Zuweisung der Größe des Speicherplatzes für diese Werkzeuge befinden sich im Programmcode einerseits am Anfang der Methode *WriteCompressedDataMLT* innerhalb der Klasse *JpegEncoder*, andererseits in der Klasse *class MLT*.
Die *x-history* besitzt eine Doppelfunktion; anfangs wird sie benutzt, um die vorzubereitenden Daten temporär zwischen zu speichern, später ist sie das zentrale Zwischenspeichermodul zur Berechnung der überlappenden Transformationen. Ihr Speichervolumen beträgt vier mal acht Werte bei einem Abtastverhältnis von 4:4:4. Hier stehen auch jeweils die letzten Werte aus dem vorangegangenen Block. Für die Berechnung der LDT und LHT verdoppelt sich die Größe der *x-history*, da die Ergebnisse der MLT′ parallel gespeichert werden müssen. Die *y-history* stellt Speicher für vier vollständige Bildzeilen zur Verfügung, die sich für die Berechnung der LDT wieder verdoppelt und für die LHT sogar vervierfacht hat. In ihr stehen die transformierten Daten von immer vier kompletten Bildzeilen. Damit diese Speicherbausteine gefüllt werden, bedarf es eines Zulieferers, der sich die Daten holt und weitergibt. Das regelt der *workbuf*, der eine Größe von acht Werten hat und bei der Berechnung der MLT′ Platz für weitere acht Werte besitzt.

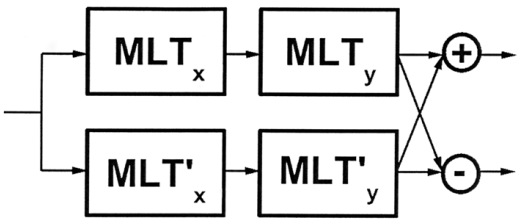


Abb. 3.2: *Blockdiagramm der LDT*

Für die Berechnung der MLT wird zunächst eine Transformation in horizontaler Richtung durchgeführt, dann in vertikaler. Ein fertig transformierter Block wird zum Schluß in den *workspace* geschrieben, der mit 64 Koeffizienten gefüllt ist. Diese werden dann einem dreidimensionalen Feld *mltArray1* zur weiteren Verarbeitung an den Quantisierer übergeben.
Der *workspace* muß für die LDT wieder vergrößert werden, auf Grund der zusätzlichen MLT′ Werte. Aus diesen wird das LDT Spektrum gemäß der Summen- und Differenzbildung in Abbildung 3.2 gebildet und wegen der Speicherung der dop-

pelten Anzahl der Koeffizienten ist das *mltArray1* auch dopplet so groß, so daß es zu zwei separaten outStreams kommt, die hintereinander gespeichert werden.

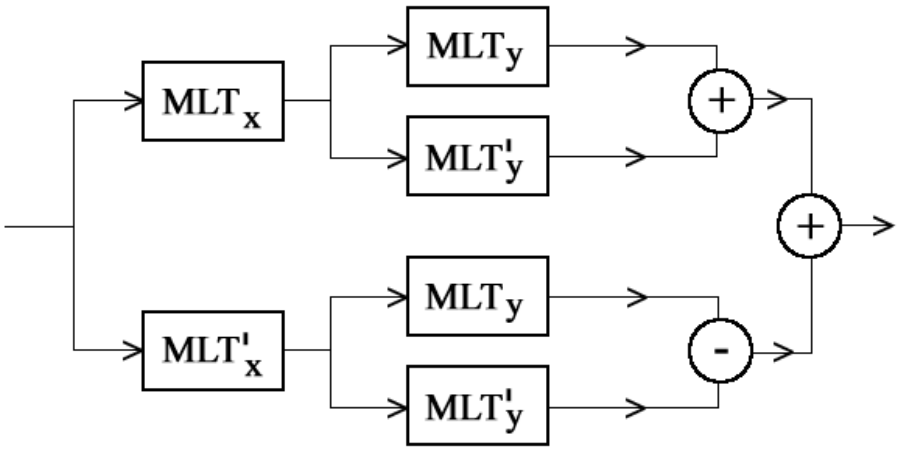


Abb. 3.3: *Das LHT Blockdiagramm*

Die LHT verzweigt sich nach der ersten horizontalen Transformation nochmals in zwei Vertikale, was eine Vervierfachung des *workspaces* zur Folge hat. Aus diesen vier Teiltransformationen wird mit entsprechender Summen- und Differenzbildung nur ein Spektrum gebildet, somit wird nur ein *mltArray1* benötigt, analog zur MLT.
Bei der Bildung der MLT′ liegt der Unterschied zur MLT im Vorzeichen; für alle ungeraden Frequenzen wird deshalb eine bitweise UND Operation durchgeführt, so daß die Werte nun positiv sind. Danach können die Werte aus dem *workspace* nach $v = 0,5 \cdot (x1 + x2 + x3 - x4)$ weiterbehandelt werden. Die Werte müssen am Ende mit 0,25 multipliziert werden, um die ursprüngliche Farb- und Helligkeitsinformation nicht zu verfälschen. Dies wird jeweils seitens des En- und Dekoders mit dem Fakor 0,5 realisiert.

3.1.2 Die LHT Vorfilterung

Wie schon erwähnt, sind die LHT Basisfunktionen nicht vollständig orthogonal. Es entstehen an Diagonalen und Ecken Geisterbilder, die sich als Umrandungsstrukturen bemerkbar machen. Dabei treten diese besonders deutlich hervor, wenn ein hoher Kontrast im Bild auftritt, wie z.B. in Abbildung 3.4 und 3.5 zwischen dem Buchstaben R zum Hintergrund.

Abb. 3.4: *ohne Filterung*

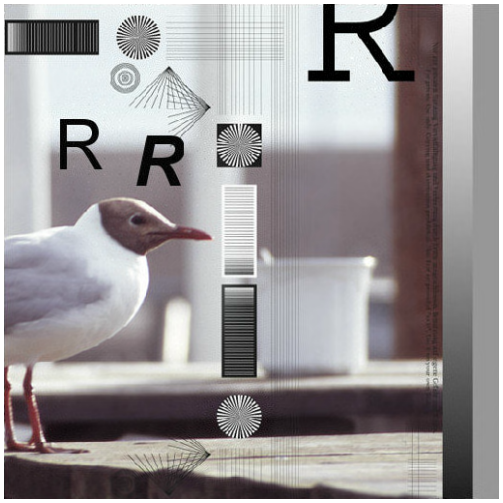


Abb. 3.5: *mit Filterung*

Die Hintransformation und die anschließende Rücktransformation ergeben also nicht das Originalbild wieder. Die Diagonalen von links oben nach rechts unten erzeugen eine negative Störung und analog dazu die von rechts oben nach links unten eine positive Störung, da das Originalbild mit einem Signal gefaltet wird, das durch einen Filterkern beschrieben werden kann, der so konzipiert ist, daß er eine 16 mal 16 Matrix beschreibt, welche in ihren „Ecken“ je einmal 0,25 und -0,25 stehen hat, das wiederum durch eine entprechende Funktion der Form $\cos(x + y) - \cos(x - y)$ beschrieben werden kann. Das Ergebnis wird dann noch mit dem cos^2-Fenster multipliziert.
Zurückzuführen ist dies auf die Zusammensetzung eines LHT Punktes aus vier MLT Teilpunkten. Diese haben zueinander in horizontaler und vertikaler Richtung eine Phasenverschiebung von $\frac{\pi}{2}$, jedoch innerhalb der Diagonalen kommt es zu einer Phasenverschiebung von π. Deshalb ist keine Orthogonalität gegeben.
Das Originalbild wird mit einer geometrischen Reihenentwicklung gefiltert, d.h. mit der Filterfunktion gefaltet. Stellt man sich dazu das Bild v als einen n-dimensionalen Vektor vor, kann man es mit der Einheitsmatrix E multiplizieren, ohne es zu verfälschen. Bedingt durch den LHT Transformationskern, ensteht aber nun eine zusätzliche Matrix A, die sich auf das Originalbild addiert.

$$E \cdot v + A \cdot v = (E + A) \cdot v \tag{3.1}$$

Das Ergebnis mit ihrer inversen Matrix multipliziert, sollte das Originalbild wieder ergeben.

$$(E + A)^{-1} \cdot (E + A) \cdot v = E \cdot v = v \tag{3.2}$$

Die inverse Matrix besteht nun aus dieser Reihenentwicklung, welche die Matrix A verschwinden läßt.

$$(E + A)^{-1} = E - A + A^2 - A^3 + A^4 ... A^N \tag{3.3}$$

Wie oft dies gemacht wird, ist im Quelltext mit dem Iterationswert *iter* einzustellen, der auf 12 gesetzt ist, weil hier der beste PSNR-Wert erreicht wird. Eine zu häufige Iteration hat dann genau den gegenteiligen Effekt; das PSNR wird schlechter und im Bild entsteht an den Stellen „ein Krisseln“, wo die Geisterbilder stark zu sehen waren.

3.1.3 Die LHT im Dekoder

Der JPEGDecoder besteht aus der äußerern Klasse *class JPEGDecoder*, mit den inneren Klassen *class ComponentSpec*, *class FrameHeader*, *class ScanComponent*, *class ScanHeader*, *class QuantisationTable*, *class HuffmanTable* und *class MLT*.
Die Methode *HuffmanValue* in der Klasse *class HuffmanTable* wurde von Professor Kunz durch einen bitBufferIndex verbessert, da einige Bilder nicht vollständig wiedergegeben wurden oder es zu einer *ArrayIndexOutOfBoundsException* bzw. *ParserDecode* Fehlermeldung[7] kam.
Die MLT, LDT und LHT nutzen den Programmcode überwiegend gemeinsam. Auf Grund der doppelten Anzahl der Koeffizienten der LDT, erfordert die Rücktransformation einen zusätzlichen Zwischenpuffer *outputBuffer*, der die Dimension Bildbreite mal doppelter MCU Höhe hat, da die Werteausgabe auf die Blockgröße von acht mal acht Pixeln ausgelegt ist.
Die LHT Rücktransformation gestaltet sich wesentlich einfacher als beim Enkoder. Hauptsächlich brauchten nur die erforderlichen Programmsegmente in die

[7] Diese Fehler treten auch bei Bildern auf, deren Pixelmaße keine Vielfache von acht sind.

Klasse *class MLT* hinzugefügt werden. Die dazu benötigten Vergrößerungen der Speicherzuweisungen mußten notwendigerweise erfolgen.
Die Daten werden zunächst aus dem *IMLT_Source* Feld in den workspace für die inverse MLT eingelesen und in den *workspace + DCTSIZE* für die inverse MLT′. Hier muß aber das Vorzeichen wieder herumgedreht werden, da für die MLT′ entgegengesetzte Symmetrieeigenschaften gelten. Dies geschieht durch eine *left shift* Operation als Vorfaktor s. Danach werden die Daten zunächst vertikal, anschließend horizontal rücktransformiert. Nach der vertikalen inversen Transformation liegen die Daten im workspace vor. Hier wird durch ein zusätzliches *Offset* Speicherplatz für dieselben MLT oder MLT′ Daten zur Verfügung gestellt. Der Trick beim Dekodieren besteht darin, die Summenbildung der rücktransformierten LHT Werte auf zwei Stufen zu verteilen, so daß kein weiterer Speicher benötigt wird, als ohnehin für die Dekodierung der MLT und LDT vorgesehen. So wird die erste Addition bzw. Differenz beim Füllen des *workbuf*s gebildet, der die Daten dann zur horizontalen inversen Transformation weitergibt. Bei der Differenzberechnung werden die MLT′ Werte wieder multipliziert mit dem Faktor s. Bei der Ausgabe der Daten vom *workbuf* in den *outputBuffer*, werden die Daten nochmals addiert und mit 0,5 wieder multiplizert.
Somit liegen die ersten rücktansformierten Werte vor, die nun in den RGB Farbraum zurück konvertiert werden müssen, bevor eine Darstellung auf dem Bildschirm erfolgen kann.

3.2 Die ImageJ PlugIns und Macros

Die LHT soll auf statistische Merkmale untersucht werden. Dazu ist es wichtig, sich die transformierten Bildaten der LHT in einem separaten Spektrumsbild ausgeben zu lassen. Dies kann im Auswahlfenster nach Aufruf des PlugIns *JPEG_Writer_* vorgenommen werden. Dabei spielt nur die Wahl des Abtastverhältnisses eine Rolle, nicht aber die Wahl der Qualitätswerte, da die Daten zu diesem Zeitpunkt den Quantisierer noch nicht durchlaufen haben. Die zu betrachtende Komponente (Y, Cb oder Cr) kann einzeln angewählt werden.

3.2.1 Var_.java

Die Berechunung der Varianz, Kovarianz und Korrelation ist in einem Quellcode *Var_.java* von Professor Kunz implementiert worden.
Digitale Daten können als diskrete Zufallsvariablen aufgefasst werden, da sie immer fehlerbehaftet sind in Form von digitalem Rauschen[8]. Die Wahrscheinlichkeitsrechnung (Stochastik) und die mathematische Statistik erlauben es, in solch zufälligen Ereignissen bestimmte Gesetzmäßigkeiten zu erkennen und damit diese zu beschreiben [18]. Ein Histogramm gibt den Wert eines Pixels zu seiner relativen Häufigkeit im gesamten Bild wieder. Damit wird abgeschätzt, mit welcher Wahrscheinlichkeitsdichte ein Pixel einen bestimmten Wert annimmt.
Neben der Häufigkeit, gibt es aber noch andere statistische Größen, die eine Zufallsvarible X charkterisieren. Der Mittelwert einer Zufallsvariable ist der Wert, der im Durchschnitt erwartet werden kann.

$$EX = \sum_{i=1}^{N} x_i \cdot p_i \tag{3.4}$$

Die Varianz ist ein Maß für die Abweichung der Zufallsvariable vom Erwartungswert. Sie ist definiert als die quadrierte Differenz jedes einzelnen Wertes zum Erwartungswert μ, und der Bildung des Erwartungswerts hierüber [18].

$$Var(X) = E(X - \mu)^2 \tag{3.5}$$

In empirischen Auswertungen sind alle statistische Größen selbst Schwankungen unterworfen. Für die Schätzung dieser Kenngrößen verwendet man den Mittelwert $\overline{x}$ und die empirische Varinaz $\overline{S}$.

$$\overline{x} = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{3.6}$$

In dem Programmcode werden die statistischen Zusammenhänge zwischen den unmittelbaren Pixelnachbarn untersucht. Dazu benötigt man den Mittelwert aller korrespondierenden Werte, d.h. es werden alle DC Werte aufsummiert und

[8] zufällige Störprozesse unterschiedlichster Arten und Auswirkungen

durch die Anzahl der acht mal acht Blöcke im Bild geteilt (entsprechend für alle AC Werte). Damit erhält man 64 Mittelwerte, aus denen dann die Varianz dieser Werte berechnet wird. Für die Berechnung der Varinaz von den DC und AC Werten gelten andere Schätzfunktionen. Da der Erwartungswert der AC Koeffizienten bekannt ist, $E_{AC}X = 0$, für die DC Werte dies aber nicht gilt, lautet der richtige „Schätzer“ für die Varianz im letzten Fall

$$\begin{aligned} \overline{S} &= \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2 = \\ &= \frac{1}{N-1}\sum_{i=1}^{N}x_i^2 - \frac{2}{N-1}\sum_{i=1}^{N}x_i \cdot \overline{x} + \frac{1}{N-1}\sum_{i=1}^{N}N \cdot \overline{x}^2 \ , \end{aligned} \tag{3.7}$$

und im Fall der Berchnung aus den AC Koeffizienten lautet dieser

$$\overline{S} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2 \ . \tag{3.8}$$

Die Varianzwerte werden in ImageJ in einem acht mal acht Pixel großen Bild ausgegeben. Um die numerischen Werte zu sehen, kann das Bild als *Text Image* gespeichert werden.

Wenn zwei oder mehrere Zufallsvariablen, z.B. die Pixelwerte von unterschiedlichen Bildern, unabhängig sind, haben sie statistisch nichts miteinander zu tun. Ein großer Teil der Statistik besteht darin, zu untersuchen, ob bestimmte Größen unabhängige Zufallsvariablen darstellen, da statistische Abhängigkeit ein Hinweis auf ursächliche Zusammenhänge ist [18]. Eine Abhängigkeit kann in der Bildverarbeitung oder allgemein in der Datenverarbeitung zur Redundanzverminderung genutzt werden und somit stärker komprimiert werden. Dies führt zu dem Begriff der Kovarianz und der Korrelation. Die Kovarianz gibt an, in welchem Maß die Fluktuationen zweier Zufallsvariablen zueinander in Beziehung stehen [3]. Sie mißt die gemeinsame lineare Streuung der Merkmale X und Y [13] und formuliert sich zu

$$Cov(X,Y) = E((X - \mu_x) \cdot (Y - \mu_y)) \ . \tag{3.9}$$

Die Kovarianz mit sich selbst entspricht der Varianz. Für die Schätzung der Kovarianz schreibt sich die Formel um zu

$$\begin{aligned} Cov(x,y) &= \frac{1}{N-1}\sum_{i=1}^{N}[(x_i-\overline{x})\cdot(y_i-\overline{y})] = \\ &= \frac{1}{N-1}\sum_{i=1}^{N}[x_i\cdot y_i - N\cdot\overline{x}\cdot\overline{y}] \ . \end{aligned} \tag{3.10}$$

Veranschaulicht man diese Streuung graphisch in einem Diagramm, erkennt man die Unkorreliertheit[9] der Werte bei Entstehung einer Wolke. Ergibt sich aber ein linearer Zusammenhang in Form einer Trendlinie, so stehen die Werte miteinander in Korrelation.
Für die Kovarianz betrachtet man die nächsten Nachbarn, das bedeutet, insgesammt sind vier Fälle[10] zu unterscheiden: von einem Pixel ist der Nächste acht Pixel in horizontaler Richtung neben ihm, der Zweite acht Pixel vertikal unter ihm, der Dritte acht Pixel in der 45° Grad Diagonalen rechts darunter und der Vierte acht Pixel in der 45° Grad Diagonalen links darunter. An den Rändern sind nicht alle Nachbarn vorhanden.
Mit Betätigen des PlugIns *Var_* werden zusätzlich zum Varinzbild zwei weitere Bilder mitausgegeben, nämlich das Kovarianzbild und das Korrelationsbild. Diese sind viermal so groß wie das Varianzbild, da die Zahlen für jeden Nachbarn ausgegeben werden. Der Korrelationskoeffizient setzt die Kovarianz der Zufallsgrößen zu den entsprechenden Varianzen in Beziehung.

$$\rho(X,Y) = \frac{Cov(X,Y)}{\sqrt{Var(X)}\cdot\sqrt{Var(Y)}} \tag{3.11}$$

Je kleiner die Zahl, desto weniger sind die Werte miteinander korreliert. Dabei ist es egal, ob die Werte negativ oder postiv sind. Es ergibt sich eine Maßzahl für die Stärke und Richtung eines linearen Zusammenhangs von Merkmalen [13].

[9] der errechnete Wert ist null
[10] siehe Abbildung 4.18, S. 38

3.2.2 PSNR_.java

Für das *Spitzen-Signal-Rausch Verhältnis* (PSNR) konnte das Programmstück aus [10] verwendet werden. Das PSNR gibt an, wie sich das Leistungsspektrum des Originals zu dem Leistungsspektrum des gestörten Bildes verhält. Ein großer Wert spricht für weniger Rauschen im Bild zum Orginal. Im nächsten Kapitel wird dieser Zusammenhang noch detaillierter vertieft.

3.2.3 ImageJ Macros

Um die Auswertung etwas vereinfachend zu automatisieren, wurden in ImageJ *Macros* erstellt. So konnten die bereits existierenden Macros *OpenMacro* und *WriteMacro* aus der Arbeit von Sylvia Schmitt [10] weiter benutzt werden. Zusätzlich wurde das Macro *Differenz* geschrieben, welches das Differenzbild des kodierten Bildes zum Original TIF als 32-bit Result[11] erstellt, mit anschließender PSNR Berechnung. Die Ergebnisse wurden mit *Microsft Excel* für jedes Bild ausgewertet und in Tabellen und Graphen zusammengefasst. Diese befinden sich alle auf der beiliegenden CD.
Mit *Corr_Macro* werden die Korrelationswerte für jede Farbkomponente hintereinander errechnet. Dazu müssen die zu betrachtenden Bilder umbenannt und mit einer laufenden Nummer versehen werden. Die Anzahl der verwendeten Bilder, sowie das Abtastverhältnis muß aktualisiert werden. Die Bilder werden dann hintereinander als *Bild+i.tif* gelesen und verarbeitet. Das PlugIn gibt immer die drei Bilder *Varianz*, *Kovarianz* und *Korrelation* aus; gespeichert wird aber nur das jeweilige Korrelationsbild. Diese werden dann in Form eines *stack* in ImageJ aufsummiert, durch die Anzahl der verwendeten Bilder geteilt und das Ergebniss als *Text Image* gespeichert.

[11] Im Ergebnis sind sowohl die positiven als auch die negativen Differenzen enthalten.

4 Auswertung und Diskussion der Ergebnisse

4.1 Bildbewertung

Im folgenden soll nun betrachtet werden, wie gut die unterschiedlichen Verfahren arbeiten. Es stellt sich anfänglich die Frage, wie ein Bild bewertet werden kann, welche Beurteilungskriterien für die Güte eines Bildes herangezogen werden können. Man unterscheidet einmal in subjektive Verfahrensmodelle („durch den Menschen“) und objektive Verfahrensmodelle („durch die Maschine“).
Die Beurteilung mit dem menschlichen Auge ist das Naheliegenste. In Sehtests kann der Proband das Originalbild mit dem veränderten Bild vergleichen und die Qualität anhand einer Skala bewerten. Die Maschine liefert nach mathematischen Berechnungen sehr präzise Ergebnisse, denen es einer richtigen Interpretation bedarf. Dabei werden Werte erfasst, welche das Auge auf Grund des begenzten Auflösungsvermögens oder der begrenzten Farbwahrnehmung nicht sehen kann. Ein Test kommt aber ohne eine Bewertung mit dem Auge nicht aus.
Deshalb empfiehlt es sich für die Qualitätsbeurteilung, die Bildfehler in vernachlässigbare, auffällige und störende Artefakte einzuteilen. Außerdem sollten auch bestimmte Regionen im Bild getrennt voneinander beurteilt werden. Dies können sein Vordergrund und Hintergrund, dominante großflächige Bereiche und detailreiche feine Strukturen, schräge scharfe Kanten oder unregelmäßig verteilte Punktmuster.
Dazu wurden verschiedene Bilder kodiert und nach auffälligen Merkmalen untersucht. Die Bilder sind überwiegend als Testbilder in der Bildverarbeitung bekannt, wie z.B. die Bilder *Baboon*, *Peppers*, *Lena*, der *See* und das s/w Bild *Boats*. Des weiteren wurden selbsterstellte Bilder, wie *MöweDet* und *Watt* bzw. computerge-

nerierte Bilder, wie *Blocks*, *Fläche*, *Schachbrett* und *Siemensstern*, zu Testzwecken verwendet.
Die Auswertung wurde in 4er Schritten vorgenommen, d.h. es wird jede vierte Qualitätsstufe von 0 beginnend bis 100 betrachtet, rechnerisch ausgewertet und graphisch visualisiert.

4.1.1 PSNR und KF

Ein geeignetes Qualitätsmaß für die Bildbewertung ist das *Peak-Signal-To-Noise Ratio* (PSNR). Es beschreibt das Verhältnis von dem maximalen Spitzenwert des Bildes zum Quadrat, zu dem *mittleren quadratischen Rekonstruktionsfehler* (MSE) und wird in Dezibel dB angegeben.

$$PSNR = 10 \log_{10} \cdot \left(\frac{x^2}{MSE} \right) [dB]. \tag{4.1}$$

Im Zähler steht anders als beim *Signal-Rausch-Verhältnis* nicht die Signalvarianz, sondern der Spitzenwert x zum Quadrat, der bei einem 8-bit kodierten Signal, den Wert 255 nicht übersteigen kann. Das *Spitzen-Signal-Rausch Verhältnis* ist der Erwartungswert einer Schätzfunktion [14].
Der *Mittlere quadratische Fehler* (MSE) wird für mehrere Berechnungen gebraucht, da er den Grad der Verzerrung zum Original angibt, also größere Differenzen fallen stärker ins Gewicht als kleinere.

$$MSE = \frac{1}{N} \cdot \sum_{i=1}^{N} (x_i - \bar{x_i})^2 \tag{4.2}$$

In den Graphen der Meßergebnisse wird der PSNR Wert gegenüber der Dateigröße halblogarithmisch aufgetragen.
Der zweite Graph verdeutlicht jeweils den Zusammenhang zwischen *PSNR* und *Kompressionsfaktor* (KF), der sich aus der Division von Originaldateigröße zur komprimierten Dateigröße ergibt.

$$KF = \frac{Originaldatei\ [KB]}{komprimierteDatei\ [KB]} \tag{4.3}$$

Des weiteren wurde der KF über dem PSNR jeweils von einem Dateiformat und allen Abtastverhältnissen in ein Diagrammm eingetragen, um zu verdeutlichen, wie die Kompressionen von einer Transformation zum Abtastverhältnis der gleichen Transformation arbeitet.
Gemeinsam ist den Verfahren über alle Abtastverhältnisse hinweg, daß mit steigendem Kompressionsfaktor nicht nur die Dateigröße kleiner wird, sondern auch das PSNR abnimmt. Ein geringeres PSNR geht mit einer schlechtern Bildqualität gleich. Die stärkste Komprimierung wird bei der 4:2:0 Unterabtastung[1] erziehlt, da sich hier die Farbinformation horizontal und vertikal um den Faktor zwei verringert.

	JPEG	**LHT**	**LDT**	**MLT**	**JPEG**	**LHT**	**LDT**	**MLT**	
444	370	4900	2400	1300	310	3100	3150	1050	**444**
422	300	3500	1650	900	240	2100	2150	1450	**422**
411	250	2700	1250	700	200	1450	1500	1100	**411**
420	250	2700	1250	700	200	1600	1650	1100	**420**

Tab. 4.1: *Verarbeitungszeit in [ms]; links Enkodierung, rechts Dekodierung*

Hinsichtlich der Verarbeitungszeit kann aus Tabelle 4.1 entnommen werden, daß mit einem Farbwert reduzierenden Abtastverhältnis sich die Zeit verkürzt (von 4:4:4 auf 4:2:0 um den Faktor 1,8) im Vergleich ohne unterabzutasten, da weniger Transformationsberechnungen durchgeführt werden müssen. Bei der LHT richtet sich zusätzlich die Dauer nach der Vorfilterung (siehe Tabelle 4.2, S. 34). Gegenüber den anderen Verfahren benötigt die LHT für die Komprimierung wesentlich mehr Zeit[2], bei der Dekomprimierung fällt der Unterschied zur LDT und MLT nicht so stark ins Gewicht. Die Zeitmessung ist hier eine relative Angabe[3], die im Verhältnis gesehen eine qualitative Aussage liefert. Ferner sind die angegebenen Zeiten in obiger Tabelle relative Durchschnittswerte von allen Bildern.
Eine eindeutige Aussage über die Klasse der Kompression kann nur in Verbindung mit der einhergehenden Dateigröße getroffen werden, da es schließlich das Ziel

[1] Die 4:1:1 Unterabtastung liefert geringfügig mehr Daten; in bestimmten Fällen auch weniger.
[2] ca. das 11-fache gegenüber JPEG
[3] Sie hängt in diesem Programm von der Auslastung des Arbeitsspeichers ab.

ist, eine möglichst kleine Datenrate zu erlangen. JPEG, MLT und LHT liefern in kleinen Abweichungen gleiche Ergebnisse, die LDT hingegen produziert ungefähr das 1,8fache der Dateigröße. Die Kurven zeigen deutlich, daß sie sich von den anderen Verfahren abhebt.

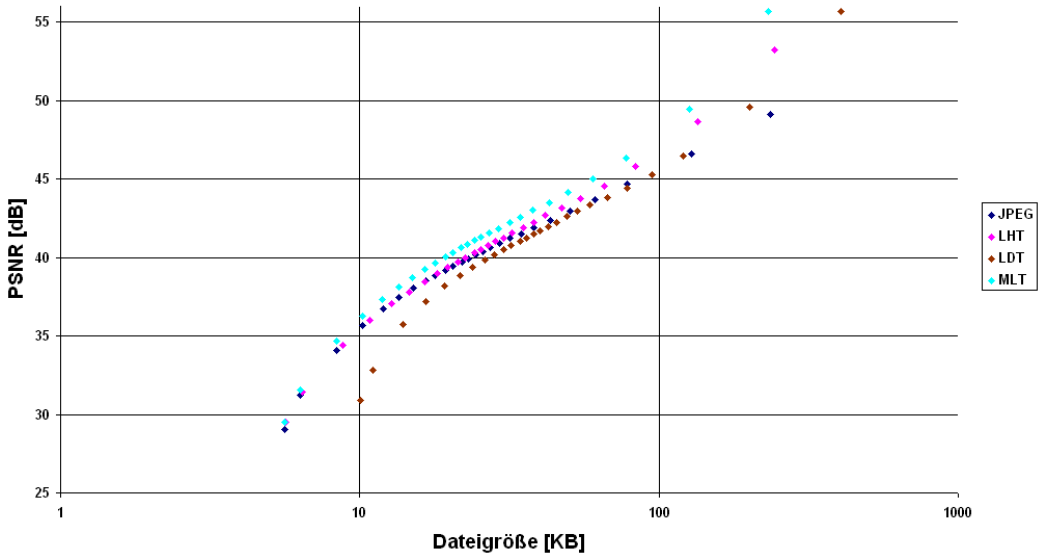


Abb. 4.1: *PSNR zur Dateigröße (logarithmisch) vom Bild 420Lena*

Das beste meßtechnische PSNR Ergebnis liefert über alle Abtastverhältnisse und Kompressionstufen hinweg die MLT. In geringen Kompressionen weicht der PSNR Wert der LHT, sowie der MLT, signifikant von dem der DCT ab, dagegen sind in starken Kompressionen nur geringfügige Unterschiede feststellbar. JPEG hat ein niedrigeres PSNR gegenüber der LHT und MLT, dafür einen leicht höheren Kompressionsfaktor. Die LDT bleibt von allen Verfahren in Bezug auf Dateigröße an letzter Stelle, jedoch vom PSNR Wert schneidet sie wiederum sehr gut ab.
An dieser Stelle sei vermerkt, daß der PSNR Wert des JPEG Verfahrens wesentlich höhere Ergebnisse erziehlt, vor allem in den hohen Qualitätsstufen, wenn nicht der erstellte Dekoder verwendet wird, sondern der von ImageJ standardmäßig integrierte Dekoder benutzt wird. Der Grund dafür läßt sich nicht definitiv belegen. Für die Auswertung ist dies aber ohne Bedeutung, da der Vergleich mit denselben Instrumenten durchgeführt werden muß.
Vergleicht man die Verfahren untereinander und trägt die Abtastverhältnisse der einzelnen Transformationen gegenüber dem Kompressionsfaktor in ein Diagramm,

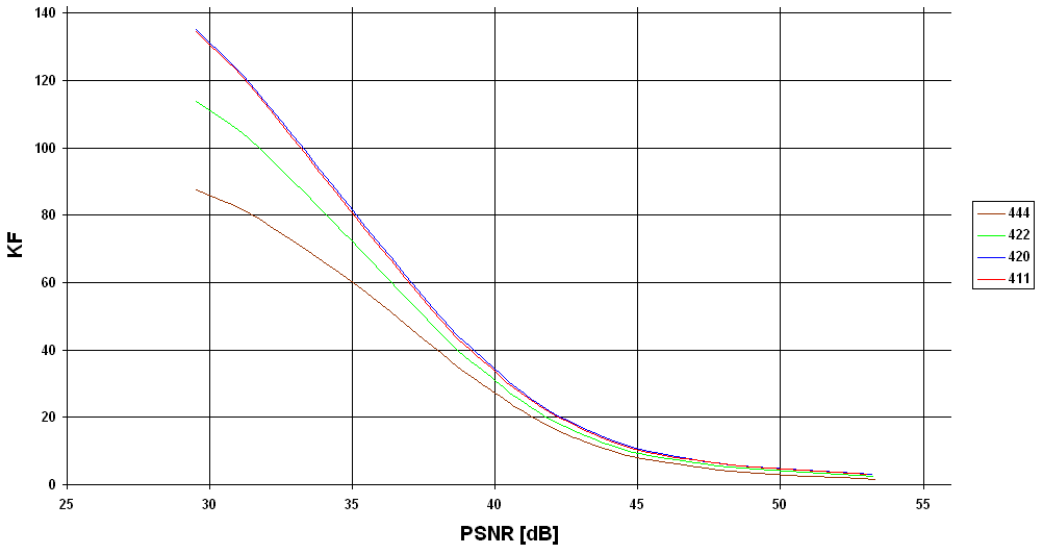


Abb. 4.2: *Abtastverhältnisse LHT vom Bild Lena*

so zeigt sich, daß die Kurven dicht beieinander liegen und die PSNR Werte durchgängig erreicht werden. Das 4:2:0 Verhältnis hat nur geringfügig abweichende PSNR Werte zu denen von 4:4:4. Alles spricht also für eine Unterabtastung, da hier die Dateigröße in kleinen Schwankungen um ein Drittel mehr reduziert ist als ohne. Betrachtet man die Werte mehrerer Bilder vom selben Typ, so ergeben sich ebenfalls ähnliche Werte im PSNR, so daß sich bei der LHT über alle Abtastverhältnisse hinweg, ein PSNR Dynamikumfang von ca. 30 [dB] ergibt.
Alle Kurven der Graphen *PSNR zur Dateigröße* lassen eine leichte *S*-Form mit einem großen „geradlinigen“ Teil erkennen. Die Werte können deshalb in drei Bereiche gegliedert werden: einer *sehr starken Kompriemierung*, eines großen *effektiven Bereichs* und einer *geringen Kompression*. Gemeint ist damit eine grobe Unterteilung der Qualitätswerte.

4.1.2 Visuelle Betrachtung

Die Bilder lassen sich in zwei Gruppen einteilen: zum einen in sehr flächige Bilder und zum anderen in sehr deteilreiche Bilder. Letztere sind mit einer vollständigen Szene ausgestattet (*Baboon*, *Peppers*, *Lena*, *See*, *Boats*, *Watt*, *MöweDet*), wohingegen die anderen nur eine Anordung von großen, geometrischen Formen darstellen (*Blocks*, *Fläche*, *Schachbrett*, *Siemensstern*). Jedes dieser Bilder zeigt

für sich die Unterschiede der Verfahren sehr deutlich. Damit formuliert sich die erste Aussage, daß bestimmte Artefakte in bestimmten Bildern besonders deutlich zu sehen sind, die Stärke des Auftretens der Fehler aber vom Bildinhalt abhängt.

Abb. 4.3: *422 Boats q=8 LHT*

Abb. 4.4: *422 Baboon q=8 LHT*

Im Bild *Baboon* sind scheinbar keine Störungen enthalten, hingegen treten bei *Boats* in derselben Kompressionsstufe deutliche Umrandungseffekte auf. Die Fehler gehen in dem stark strukturierten Affen unter und werden erst in noch stärkeren Quantisierungen sichtbar.
In sehr starken Komprimierungen (Qualitätswerte $0 \leq q \leq 10$), werden die Effekte einer zu starken Quantsierung bei allen Verfahren extrem sichtbar (vgl. Abb. 4.5 bis 4.8). Es entstehen großflächige *Contouring* Artefakte in abgestuften Übergängen und feine Linien werden nur noch ansatzweise wiedergegeben, da die Bilder auf eine sehr geringe Anzahl von Bits reduziert sind. Die Farbinformation ist auf ein Minimum beschränkt oder fehlt vollständig. Im Bild *MöweDet* sind die kleinen Siemenssterne beim JPEG Bild in der Mitte mit homogenen Quadraten[4] bestückt, wohingegen bei den überlappenden Transformationen noch die feinen Linien gut zu erkennen sind. Das ist ein grundlegender Vorteil der Bildkodierung mit derartig ausgeweiteten Basisfunktionen über die Blockgrenze hinaus.
Die LHT gibt weiterhin diagonale Linien auf Grund ihrer stark gerichteten Basisfunktionen wesentlich genauer wieder, während bei der MLT und LDT allmählich

[4] mit einer Größe von acht mal acht Pixeln

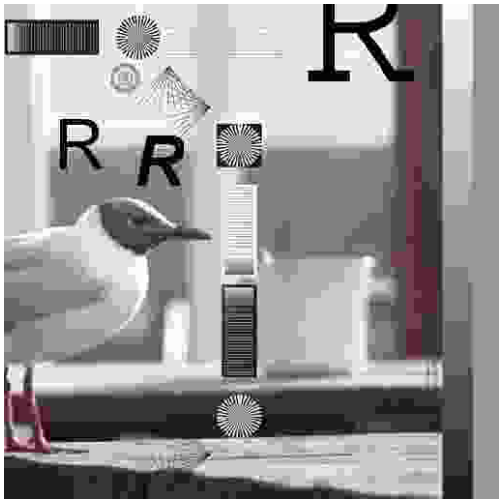


Abb. 4.5: *420 MöweDet q=4 JPG*

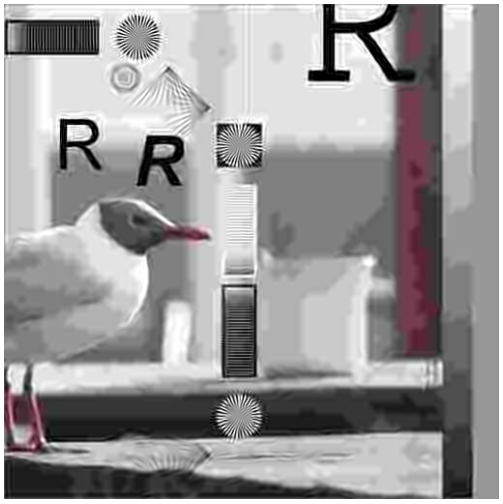


Abb. 4.6: *420 MöweDet q=4 LDT*

einheitliche Flächen zum Vorschein kommen. Im Kopf der Möwe wird die Blockbildung beim DCT basierten Verfahren deutlich sichtbar, so daß die Rundung des Kopfes nicht vollständig wiedergegeben wird, im Gegensatz zu den anderen Verfahren. Allerdings wirkt das LHT Bild durch die entstehenden Umrandungen der Geisterbilder verschmierter als das MLT oder LDT Bild. Im Vergelich zum JPEG Bild erscheinen die Bilder mit den überlappenden Transformationen leicht weichgezeichnet, was auf die Multiplikation mit dem Sinusfenster zurückzuführen ist. Die LDT gibt den geglättetsten und homogensten Bildeindruck wieder, da-

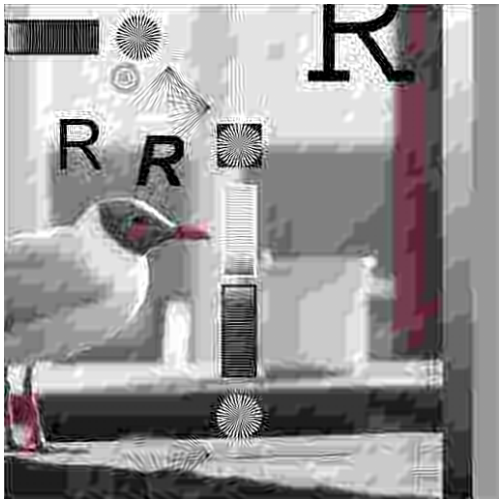


Abb. 4.7: *420 MöweDet q=4 LHT*

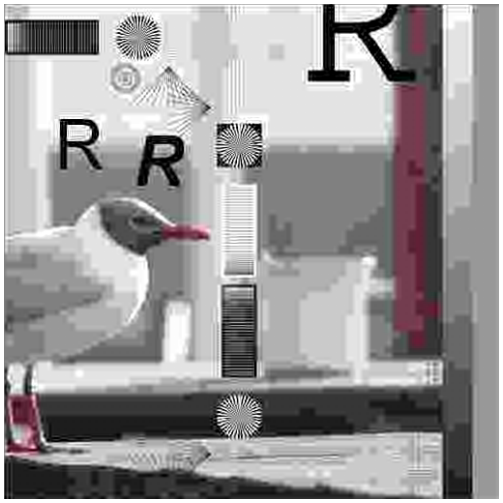


Abb. 4.8: *420 MöweDet q=4 MLT*

gegen hat die LHT sehr auffällige Bildfehler, obwohl sie prinzipiell die meisten Informationen enthält.
Das starke Krisselmuster, welches sehr deutlich um den Buchstaben R entstanden ist, rührt von der Vorfilterung des LHT Bildes her. Betrachtet man dieses mit der Lupe, so sieht man sich überlappende zwei mal zwei Pixelblöcke, die je nach Richtung entweder eine helle oder eine dunkle Information tragen. Sie treten immer dann besonders auffällig auf, wenn innerhalb eines Bildes ein sehr hoher Kontrastsprung auftritt und sich der Inhalt im Bild abrupt ändert, z.B. von einer steilen Kante zu einem flächenhaften Inhalt. Außerdem sind schräge Strukturen besonders ausschlaggebend.
In den nächsten Komprimierungsstufen (Qualitätswerte $10 \leq q \leq 80$) werden Artefakte anfangs sehr stark zu sehen sein, verschwinden aber in höheren Quälitäten zunehmend. Die Fehler wirken sich einmal als störende Effekte aus, sind dann noch auffällig wahrnehmbar und gehen schließlich in Vernachlässigbare über. Die Farbinformation wird ab q=10 schon sehr kräftig wiedergegeben.

Abb. 4.9: *420 Lena q=16 LHT*

Abb. 4.10: *420 Lena q=64 LHT*

Im LHT Bild *Lena* tritt das hochfrequente „Krisselmuster“ um den Hut, generell in langen, schräg verlaufenden Linien, als sehr unruhig und störend in Erscheinung. Dieser Effekt äußert sich in horizontalen und vertikalen Strukturen nicht in diesem Maße, jedoch können leicht verwischte Umrandungen parallel dazu entstehen. Mit steigendem Qualitätswert schwächt das Muster ab. Einige Basisstrukturen der eher tieferen Frequenzen kommen im Hut deutlich zum Vorschein, auch in

den anderen Formaten. Das LDT Bild zeigt wieder das einheitlichste Ergebnis, die parallelen Umrandungen halten sich dezent im Hintergrund und der Hut ist mit wenigen Unebenheiten versehen. Lediglich der linke Rand zeigt einen zwölf Pixel breiten farbigen Verlauf, der ohne Unterabtastung weniger auffällt. Allerdings produziert sie die 1,6fache Datenmenge. Beim JPEG Verfahren kann nahezu jeder acht mal acht Block schon in der Darstellung bei 100 Prozent erkannt werden, innerhalb deren in leichter Vergrößerung die Basisfunktionen ebenfalls hervortreten. Die MLT zeigt von jedem etwas, d.h. leichte treppenstufige Contouring Artefakte im Hintergrund, wenige Umrandungen des Hauptmotives und einige Muster der Basisfunktionen im Hut. In den MLT Bildern liegen die Fehler wesentlich minimierter vor und in Anbetracht der Dateigröße und des PSNR Wertes, steht dieses Verfahren an erster Stelle.
In den letzten 20 Qualitätsstufen können im Grunde alle Bildfehler als vernachlässigbar betrachtet werden, da sie entweder sehr unterschwellig auftreten oder nur durch eine entsprechende Vergrößerung erkannt werden.

Abb. 4.11: *420 Boats q=80 LHT*

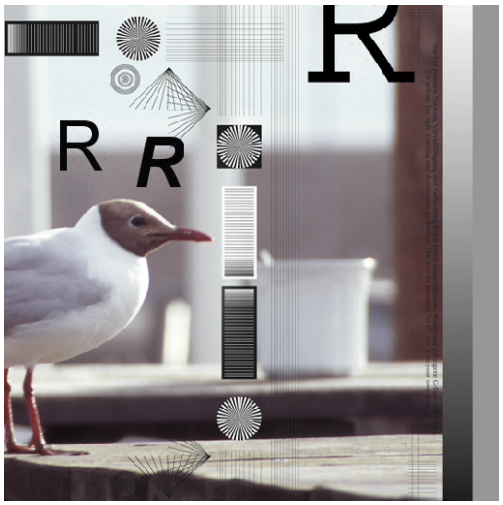


Abb. 4.12: *420 MöweDet q=100 LHT*

Im Bild *Boats* sieht man die Streifen des linken Randes erst in einer dreifachen Vergrößerung und die Umrandungseffekte in den schrägen Linien werden überhaupt nicht mehr beobachtet, sowie das Krisselmuster nur in direktem Vergleich mit den anderen Verfahren ins Auge tritt.
Beim *MöweDet* Bild macht sich ab ca. Qualitätswert 50 sukzessive ein weiterer Fehler um den Buchstaben R bemerkbar: zwei mal zwei Pixel Qaudrate, die ab-

wechselnd helle oder dunkle Information tragen und einen waagrechten Abstand von 14 Bildpunkten sowie einen Senkrechten von sechs haben. In den Bildern *Siemensstern* und *Blocks* tritt dieser Effekt wegen des hohen Kontrastes wesentlich deutlicher hervor. Das Bild *Schachbrett* zeigt einen ähnlichen Effekt in den Ecken eines dunklen Feldes zu 12 mal 12 hellen Quadraten mit horizontal und vertikal angeordneten Streifen über das gesamte Bild. Noch stärker sieht man dies ohne eine Vorfilterung des Signals.
Die JPEG Bilder zeigen in *MöweDet* auch in der besten Qualitätsstufe beim Dekodieren helle Streifen in und um den Buchstaben R herum. Beim Öffnen der Bilder über ImageJ direkt, werden diese korrekt ohne Streifen dargestellt. Die Ursache dafür konnte nicht bewiesen werden.

4.1.3 Der Iterationswert des LHT Vorfilters

Die Wirkungsweise der Vorfilterung soll hier noch einmal etwas genauer dargestellt werden. Auf die Frage, wie oft die LHT Vorfilterung durchlaufen werden soll, wurde bereits festgestellt, daß bei einer Iteration von zwölf, durchgängig das beste PSNR erziehlt wird.

	Zeit[ms]	**Größe [KB]**	**KF**	**PSNR**
0	4647	35,82	21,46	35,276
1	4757	37,67	20,41	38,353
2	4877	38,48	19,98	40,105
11	5648	41,17	18,67	41,497
12	5929	41,31	18,61	**41,505**
13	6027	41,42	18,56	41,495
20	6809	42,00	18,30	41,466
50	13229	42,92	17,91	41,287

Tab. 4.2: *Darstellung des PSNR Wertes mit untschiedlichen Iterationswerten vom Bild 444MöweDet q=70*

Höhere Werte bringen keine westentliche Verbesserung; das PSNR wird schlechter und das Krisseln nimmt leicht zu.
Die LHT kommt ohne die Vorfilterung nicht aus, besonders für die nicht so stark

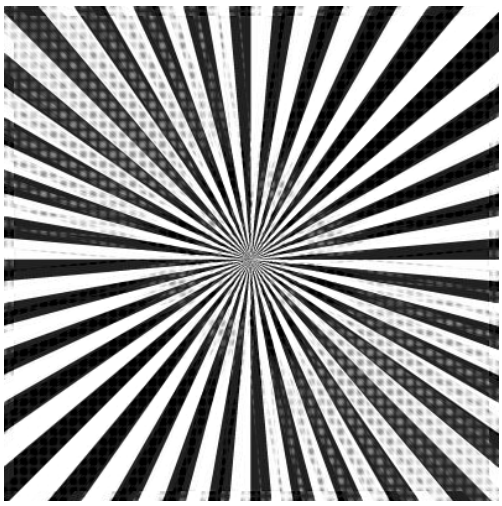


Abb. 4.13: *ohne Filterung*

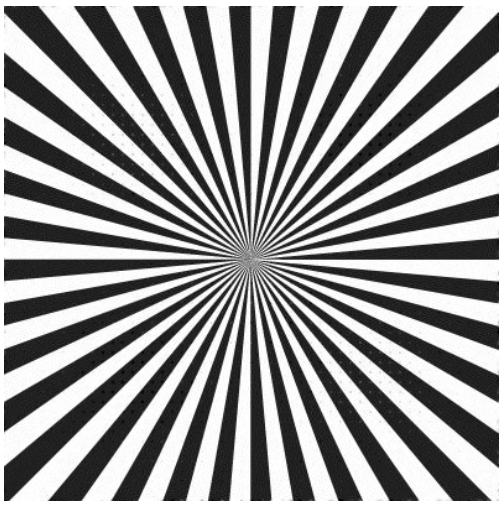


Abb. 4.14: *mit Filterung*

komprimierenden Qualitätsstufen. Das Bild *444Siemenstern q=70 LHT* veranschaulicht den Zusammenhang noch einmal.
Der Siemensstern wird normalerweise verwendet, um das Auflösungsvermögen eines optischen Systems[5] zu bestimmen. Hier soll wegen des hohen Kontrastes und den vielzahligen schrägen Linien, das Aufkommen der zwei mal zwei Pixelquadrate gezeigt werden, da der Fehler mit ungefilterten Daten größer zu sein scheint.
Allerdings sollte nicht unerwähnt bleiben, dass es bei allen Verfahren, in starken

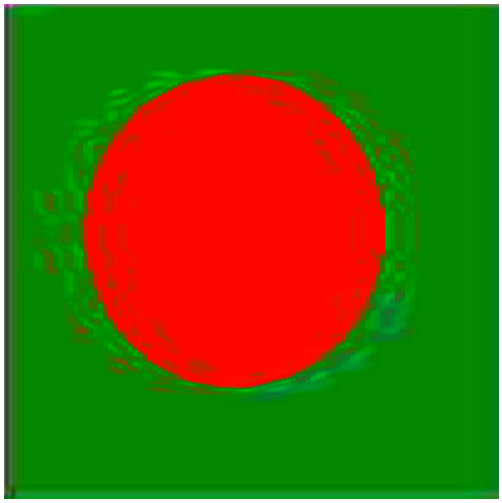


Abb. 4.15: *ohne Filterung*

Abb. 4.16: *mit Filterung*

[5] z.B. von Objektiven

Kompressionen, zu einem Übergreifen von schwarzen Bildwerten in die Weißen kommt, wegen des Wechsels zwischen den zwei Helligkeitswerten. Jedoch ist bei den LHT Bildern eine Regelmäßigkeit zu erkennen, da die Quadrate von links oben nach rechts unten ein Muster bilden und umgekehrt.
Es zeigt sich, daß die Vorfilterung in starken Kompressionen nicht zwingend eine Verbesserung bringt. Vielmehr kann es scheinbar zu keiner Veränderung oder zu „Schmier-"effekten[6] kommen, wie die Bilder aus Abbildung 4.15 und 4.16 *411Fläche q=4 LHT* demonstrieren. In diesem Fall ist das Bild mit Vorfilterung schlechter als das ohne. Das ist aber nicht die Regel.

4.1.4 Das Differenzbild

Um Bilder miteinander zu vergleichen gibt es noch weitere Möglichkeiten. Mit dem Differenzbild wird der Unterschied zwischen zwei Bildern sichtbar. Dabei ist es egal, welches Bild von welchem abgezogen wird. Das Ergebnis ist immer der gemeinsame Unterschied zwischen beiden Bildern.

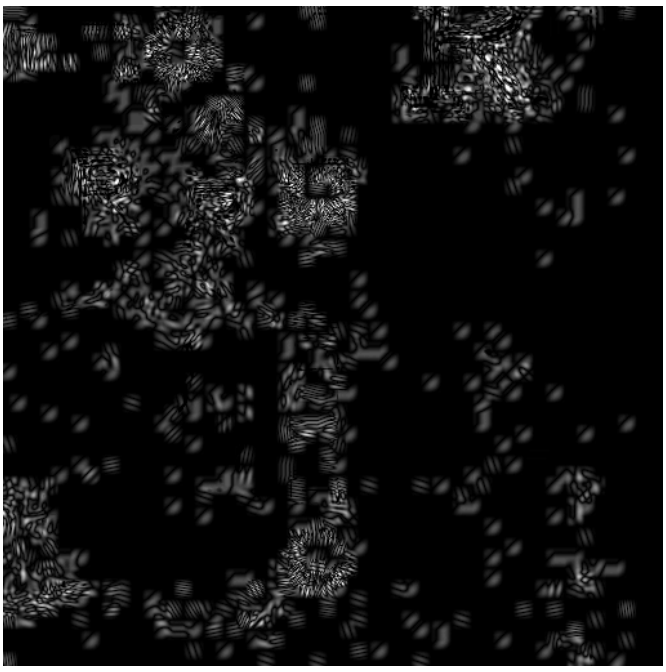


Abb. 4.17: *Differenzbild MöweDet*

[6] Im verstärktem Maße bei einer Unterabtastung der Farbwerte.

Dies wird unter anderem bei der Berechnung des PSNR Wertes genutzt. Es ist durchaus von Interesse, das Differenzbild visuell zu betrachten, da man Fehler besser lokalisieren und einordnen kann. Bildet man beispielsweise die Differenz von *420MöweDet4* ohne Filterung zu dem mit Filterung, so kommen die Stellen zum Vorschein, in denen das Krisselmuster ensteht. Damit läßt sich erkennen, welche Frequenzen offensichtlich zu Störungen führen.

4.2 Statistische Auswertung

Es scheint von größerem Interesse, die Pixelwerte genauer auf Abhängigkeiten zu untersuchen, denn stellt man derartige Zusammenhänge statistisch fest, so ist es mit einem neuen Kodierungsalgorithmus möglich, weitere redundante Informationen wegzulassen, um schließlich zu einer noch kleineren Datentrate zu gelangen. Zwei völlig unterschiedliche Bilder können z.B. die selben Histogramme[7] liefern, die Bildstruktur wird dabei aber nicht berücksichtigt. Im Fourierraum beinhaltet jeder Punkt zwei Informationen, Amplitude und Phase, d.h. die relative Position einer periodischen Struktur [3]. Die Phaseninformation ist dabei maßgeblich für die richtige Zuordnung der Bildpunkte.
Deshalb wurden mehrere Bilder in Form eines Korrelationsbildes auf mögliche Abhängigkeiten untersucht. Betrachtet wurden die Werte der LHT.

4.2.1 Varinaz, Kovarianz, Korrelation

Die Untersuchungen stützen sich auf die unmittelbaren Nachbarn innerhalb eines quadratischen Rasters. Diese ergeben sich aus Abbildung 4.18 zu vier korrespondierenden, nächsten Nachbarn.
Das Ergebnis der Korrelationswerte hat zweierlei Bedeutung: zum einem besagt es, daß die Werte untereinander abhängig oder unabhängig sind und zum anderen kann eine Aussage über die darin enthaltenen Basisfunktionen getroffen werden.
Es wurden jeweils sechs RGB Bilder[8] mit einer Größe von 512 mal 512 Pixeln für alle Abtastverhältnisse betrachtet, deren Bildinhalt sich von Portait bis Land-

[7] die Werte werden dabei nach Graustufe und Häufigkeit sortiert
[8] *MöweDet, Peppers, Watt, Lena, Baboon, See*

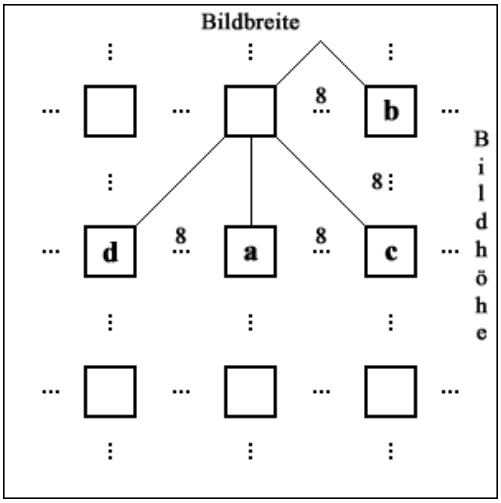


Abb. 4.18: *Die zueinandergehörigen, nächsten Pixelnachbarn*

schaftsaufnahme erstreckt. Jeweils werden vier mal 64 Werte ausgegeben, welche die Korrelationen zu den nächsten Nachbarn in der Reihenfolge b, a, c, d sind. Dies wurde einmal für die gefilterten Daten gemacht und entsprechend einmal ohne Iteration.

4.2.2 Interpretation der Werte

Die Ergebnisse für die Y Komponente sind für alle Abtastverhältnisse gleich, die für die Farbkomponenten unterschiedlich. Der erste Wert ist jeweils der größte von allen. Dies ist der DC Korrelationswert, von dem vorraus zu setzten war, daß er mit seinem Vorgänger stark korreliert ist, da bereits im Differenzverfahren die Ähnlichkeit der Grauwerte ausgenutzt wird. Er hat immer einen positiven Wert. Allerdings ist der DC Wert bei den überlappenden Transformationen nicht gleichbedeutend mit dem der DCT, da die tiefste Frequenz nicht nur die Helligkeitsinformation enthält, sondern bereits auch eine gerichtete Struktur.
Danach folgen die Zahlen der 63 AC Koeffizienten. Die Werte lassen nicht zwingend eine geringe Korrelation hochfrequenter Anteile oder eine hohe für niedrige Frequenzen erkennen. Die Richtung wird aber durch das Vorzeichen bestimmt. Im Fall b ist überwiegend jede zweite Spalte negativ, im Fall a jede zweite Zeile. Hier ist ein leichter Trend abzusehen, daß die tiefen und die hohen Frequenzen stärker miteinander korreliert sind. Der Großteil der restlichen Korrelationswer-

te sind sehr klein und fallen gegenüber den tiefen und hohen Frequenzen stark auf Null hin ab. In der ersten Diagonalen *c* sind alle Werte positiv und haben, gegenüber den anderen Beziehungen, einheitlichere AC Korrelationswerte, d.h. abgesehen von dem sehr großen DC Wert, ist der Unterschied des größten AC Korrelationswertes zum kleinsten nicht so groß, wie bei den Fällen a und b. In der anderen Richtung *d* gilt gleiches, nur sind alle Korrelationswerte (bis auf den DC Wert) negativ, mit etwas kleineren Werten als im Fall *c*.
Die Farbkomponenten zeigen ähnliches Verhalten. Die Werte werden mit steigender Farbwertreduzierung kleiner. Die 63 AC Korrelationswerte der diagonalen Nachbarn sind wertäquivalenter, als die des daneben und darunter Liegenden.
Die Vorfilterung hat keinen großen Einfluß auf diese Untersuchungen. Die Zahlen sind kleinen Schwankungen unterlaufen und sind in der Regel etwas kleiner gegenüber denen mit Filterung.

5 Schluss

Mit der Entwicklung der LHT ist eine Verkleinerung der zu speichernden Daten im Vergleich zur LDT drastisch verbessert worden und somit, in Bezug auf die Dateigröße, konkurrenzfähig zum JPEG und MLT Verfahren. Lediglich in geringen Kompressionen ist die Datenmenge geringfügig größer.
Durch die unterschiedlich gerichteten Basisfunktionen, hat die LHT in hochfrequenten Anteilen eine sehr gute Detailwiedergabe, besonders in sehr starken Kompressionen. Blockartefakte, wie beim JPEG Verfahren, treten nicht auf.
Als ein entscheidender Nachteil der LHT, ist die Entstehung der Geisterbilder. Diese werden zwar durch die Vorfilterung der Daten nahezu vollständig beseitigt, jedoch kann an kritischen Stellen ein bleibender Rest bestehen, der sich als ein Krisselmuster visuell auswirkt. Durch die Filterung verlängert sich zusätzlich die Verarbeitungszeit.
Mit der erfolgreichen Implementierung in Java, kann die LHT perfekt für vergleichende Untersuchungen der Bildkompression verwendet werden, zumal der Programmcode nach Bedarf geändert und erweitert werden kann.
Für folgende Arbeiten kann es daher von Interesse sein, den Transformationskern mathematisch kompakter zu gestalten, so daß eine Vorfilterung nicht mehr nötig ist und dadurch sich die Verarbeitungszeit verkürzt.
In Bezug auf eine statistische Vorhersage von Werten, konnten in dieser Arbeit nur skizzenhaft Ergebnisse erbracht werden und ist im Weiteren zu untersuchen. Die LHT ist eine überlappende Transformation mit hohem Potential, bei der man auf weitere Forschungsprojekte gespannt sein darf.

Literaturverzeichnis

[1] H.S. Malvar, *Signal Processing With Lapped Transforms*, UMI Books on Demand, (1992)

[2] W. Burger, M.J. Burge, *Digitale Bildverarbeitung*, Springer, (2005)

[3] B. Jähne, *Digitale Bildverarbeitung*, Springer, (2005)

[4] T. Strutz, *Bilddatenkompression*, Vieweg, (2005)

[5] R.H. Bracewell, *Schnelle Hartley-Transformation*, Oldenburg, (1990)

[6] Ph.W. Besslich, T. Lu, *Diskrete Orthogonaltransformationen*, Springer-Verlag, (1990)

[7] W. Abmayer, *Einführung in die digitale Bildverarbeitung*, B.G. Teubner, (1994)

[8] T. Aach, D. Kunz, *A lapped directional transform for spectral image analysis and its application to restoration and enhancement*, Signal Processing 80, (2000)

[9] K.E. Marquardt, *Untersuchung zur Bildkompression mit überlappenden Transformationen*, Diplomarbeit Fachbereich Photoingenieurwesen und Medientechnik Fachhochschule Köln, (2002)

[10] S. Schmitt, *Implementierung eines javabasierten Bildkompressionscodecs unter Verwendung überlappender Transformationen*, Diplomarbeit Fachbereich Photoingenieurwesen und Medientechnik Fachhochschule Köln, (2005)

[11] H. Mössenböck, *Sprechen Sie Java?*, dpunkt.verlag, (2003)

[12] G. Krüger, *Handbuch der Java-Programmierung*, Addison-Wesley, (2005)

[13] R. Mohr, *Statisik für Ingenieure und Naturwissenschaftler*, expert verlag, (2003)

[14] K. Bosch, *Großes Lehrbuch der Statistik*, R. Oldenburg, (1996)

[15] ITU/CCITT, *Information Technology - Digital Compression And Coding Of Continuous-Tone Still Images - Requirements And guidelines*, http://www.w3.org/Graphics/JPEG/itu-t81.pdf, (1992)

[16] W.B. Pennebaker, J.L. Mitchell, *Jpeg Still Image Data Compression Standard*, Van Nostrand Reinhold, (1993)

[17] D. Kunz, *Digitale Bidverarbeitung Vorlesungsskript*, http://www.fo.fh-koeln.de/dozenten/dkunz/skripte/bv/, (2002)

[18] D. Kunz, *Fouriertransformation Vorlesungsskript*, http://www.fo.fh-koeln.de/dozenten/dkunz/skripte/fourier/, (2001)

A Anhang

A.1 Farbraumtransformation und Abtastverhältnisse

Die RGB Daten werden nach folgender Formel in den YCbCr Farbraum transformiert.

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0,299 & 0,587 & 0,114 \\ -0,169 & -0,331 & 0,500 \\ 0,500 & -0,419 & -0,081 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{A.1}$$

Damit liegen die Helligkeitswerte getrennt von den Chrominanzwerten vor.
Der Wertebereich ergibt sich für die einzelnen Komponenten von

$$0 \leq Y \leq 255 \quad -127,5 \leq Cb \leq 127,5 \quad -127,5 \leq Cr \leq 127,5 \ .$$

Die Farbinformation kann horizontal und vertikal unterabgetastet werden; dadurch wird festgelegt, wieviele Data Units der jeweiligen Komponente in einer MCU zu kodieren sind.

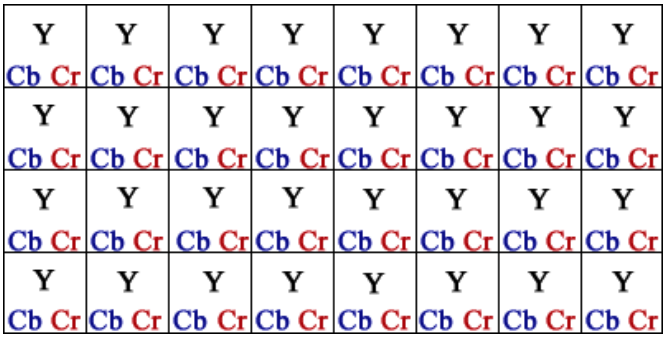


***Abb. A.1**: 4:4:4 Format; keine Unterabtastung*

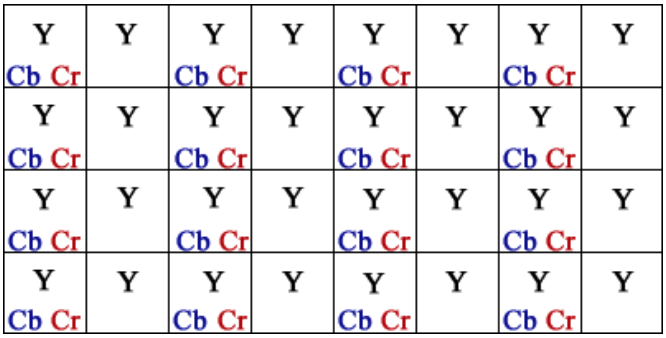


Abb. A.2: *4:2:2 Format; horizontale Unterabtastung Faktor 2*

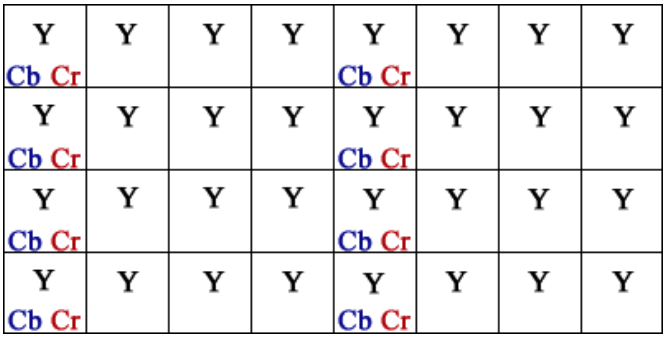


Abb. A.3: *4:1:1 Format; horizontale Unterabtastung Faktor 4*

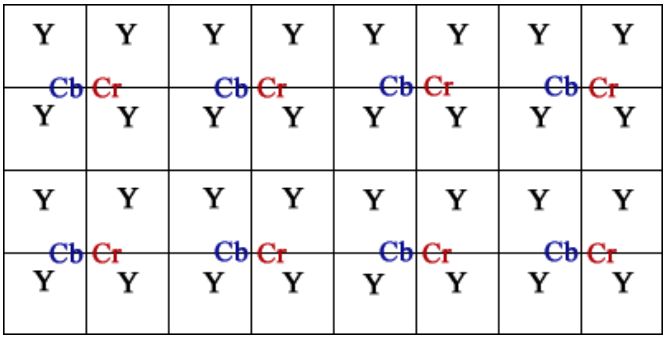


Abb. A.4: *4:2:0 Format; horizontale und vertikale Unterabtastung Faktor 2*

A.2 Ergebnisse der Korrelationswerte

Die Korrelationswerte für die Y Komponente mit Filterung bei 444:

0.856	-0.249	0.220	-0.169	0.093	-0.119	0.048	0.009
0.417	-0.276	0.219	-0.154	0.158	-0.136	0.108	-0.070
0.319	-0.209	0.216	-0.155	0.160	-0.156	0.099	-0.073
0.224	-0.047	0.097	-0.147	0.092	-0.149	0.054	-0.068
0.322	-0.127	0.098	-0.092	0.134	-0.133	0.097	-0.050
0.414	-0.103	0.094	-0.106	0.099	2.21E-4	0.023	-0.006
0.441	-0.116	0.032	-0.084	0.094	0.026	0.058	-0.138
0.474	-0.111	0.088	-0.042	0.069	-0.006	8.28E-4	-0.139
0.874	0.491	0.397	0.404	0.435	0.539	0.602	0.639
-0.214	-0.182	-0.147	-0.065	-0.047	-0.106	-0.050	-0.060
0.212	0.197	0.112	0.122	0.074	0.084	0.069	0.056
-0.125	-0.088	-0.076	-0.095	-0.099	-0.073	-0.002	-0.032
0.128	0.127	0.157	0.084	0.104	0.042	0.101	0.049
-0.107	-0.140	-0.111	-0.120	-0.033	-0.017	0.209	-0.012
0.046	0.093	0.073	0.059	0.025	0.132	0.045	0.073
-0.012	-0.058	-0.047	-0.053	-0.062	-0.110	-0.121	-0.081
0.787	0.051	0.247	0.108	0.122	0.044	0.075	0.090
0.089	0.220	0.191	0.247	0.144	0.206	0.179	0.174
0.249	0.209	0.284	0.219	0.219	0.192	0.220	0.203
0.168	0.229	0.221	0.195	0.142	0.230	0.140	0.198
0.143	0.118	0.227	0.207	0.218	0.195	0.217	0.136
0.068	0.214	0.199	0.209	0.167	0.202	0.146	0.179
0.101	0.151	0.200	0.170	0.190	0.199	0.150	0.214
0.090	0.195	0.160	0.196	0.145	0.089	0.206	0.248
0.778	-0.239	-0.091	-0.202	-0.096	-0.122	-0.083	-0.052
-0.220	-0.093	-0.292	-0.233	-0.169	-0.178	-0.178	-0.171
-0.121	-0.304	-0.284	-0.271	-0.217	-0.202	-0.177	-0.198
-0.160	-0.197	-0.249	-0.178	-0.179	-0.162	-0.217	-0.187
-0.106	-0.155	-0.176	-0.181	-0.188	-0.215	-0.225	-0.183
-0.140	-0.165	-0.198	-0.149	-0.160	-0.199	-0.204	-0.188
-0.104	-0.171	-0.197	-0.221	-0.163	-0.102	-0.096	-0.163
-0.090	-0.184	-0.181	-0.186	-0.164	-0.162	-0.115	-0.042

Tab. A.1: *Korrelationswerte 444 in Reihenfolge b, a, c, d*

Die Korrelationswerte für die Cb Komponente mit Filterung bei 444:

0.858	-0.193	0.137	-0.123	0.078	-0.093	0.023	-0.011
0.491	-0.207	0.177	-0.143	0.089	-0.057	0.051	-0.029
0.456	-0.176	0.158	-0.117	0.084	-0.053	0.046	-0.026
0.486	-0.047	0.102	-0.076	0.050	-0.059	-0.034	-0.048
0.533	-0.106	0.026	-0.034	0.078	0.014	0.070	-0.046
0.525	-0.064	0.091	-0.053	0.034	0.063	-0.031	-0.009
0.537	-0.054	0.016	-0.026	-0.028	0.079	0.026	-0.143
0.551	-0.068	0.076	-0.019	0.026	-0.009	-0.025	-0.169
0.854	0.593	0.493	0.476	0.518	0.592	0.560	0.575
-0.213	-0.196	-0.145	-0.062	-0.07	-0.067	-0.035	-0.032
0.165	0.165	0.104	0.070	0.020	0.077	0.033	0.050
-0.076	-0.132	-0.072	-0.054	-0.050	-0.003	0.055	0.028
0.077	0.100	0.074	0.039	0.075	-0.044	0.064	0.058
-0.040	-0.078	-0.026	-0.052	0.011	-0.012	0.272	0.016
0.041	0.062	0.027	0.007	-0.091	0.117	0.043	0.129
-2.98E-4	-0.012	-0.023	-0.052	-0.052	-0.146	-0.098	-0.110
0.756	0.051	0.171	0.067	0.114	-0.003	0.055	0.062
0.096	0.242	0.220	0.215	0.126	0.163	0.135	0.158
0.211	0.185	0.266	0.209	0.157	0.168	0.148	0.140
0.117	0.197	0.166	0.166	0.156	0.170	0.062	0.158
0.098	0.124	0.148	0.147	0.191	0.161	0.159	0.095
0.066	0.130	0.175	0.162	0.194	0.116	0.106	0.156
0.079	0.133	0.149	0.145	0.249	0.201	0.117	0.141
0.077	0.146	0.137	0.168	0.152	0.130	0.152	0.224
0.738	-0.214	-0.106	-0.143	-0.050	-0.131	-0.063	-0.059
-0.230	-0.120	-0.248	-0.202	-0.183	-0.138	-0.148	-0.139
-0.088	-0.246	-0.236	-0.236	-0.182	-0.159	-0.150	-0.142
-0.113	-0.173	-0.231	-0.137	-0.167	-0.159	-0.179	-0.213
-0.035	-0.166	-0.159	-0.159	-0.137	-0.234	-0.242	-0.168
-0.122	-0.152	-0.165	-0.133	-0.153	-0.141	-0.236	-0.159
-0.050	-0.160	-0.162	-0.150	-0.175	-0.059	-0.007	-0.184
-0.064	-0.152	-0.164	-0.183	-0.176	-0.130	-0.095	-0.014

Tab. A.2: *Korrelationswerte 444 in Reihenfolge b, a, c, d*

Die Korrelationswerte für die Cr Komponente mit Filterung bei 444:

0.852	-0.188	0.171	-0.123	0.056	0.011	0.042	-0.135
0.527	-0.220	0.166	-0.130	0.072	0.008	0.073	-0.057
0.433	-0.151	0.147	-0.090	0.062	-0.081	0.073	-0.030
0.476	-0.075	0.077	-0.042	0.051	-0.045	0.012	-0.035
0.497	-0.068	0.056	-0.045	0.051	-0.074	0.040	-0.026
0.458	-0.095	0.058	-0.046	0.041	0.021	-0.019	-0.028
0.454	-0.109	0.037	-0.029	0.015	0.048	0.009	-0.111
0.533	-0.067	0.050	-0.018	0.022	0.037	0.004	-0.180
0.844	0.542	0.413	0.355	0.271	0.292	0.346	0.414
-0.227	-0.229	-0.119	-0.066	-0.059	-0.166	-0.103	-0.068
0.166	0.127	0.112	0.048	0.028	0.036	0.028	0.015
-0.075	-0.104	-0.076	-0.039	-0.024	-0.006	0.016	0.019
0.100	0.062	0.054	0.031	0.051	0.024	0.063	0.029
-0.022	-0.024	-0.035	-0.041	-0.026	-0.002	0.191	0.005
0.117	0.073	0.036	0.048	-0.020	0.041	-0.068	0.080
-0.020	-0.024	-0.030	-0.023	-0.048	-0.103	-0.115	-0.120
0.737	0.070	0.180	0.115	0.142	0.044	-0.003	-0.007
0.013	0.215	0.201	0.206	0.130	0.165	0.127	0.155
0.208	0.199	0.255	0.197	0.169	0.161	0.159	0.142
0.099	0.165	0.161	0.154	0.149	0.187	0.088	0.159
0.126	0.141	0.142	0.126	0.175	0.146	0.159	0.129
0.028	0.101	0.160	0.156	0.139	0.148	0.078	0.142
0.053	0.116	0.159	0.146	0.195	0.128	0.118	0.108
0.077	0.160	0.154	0.144	0.160	0.087	0.117	0.218
0.741	-0.218	-0.099	-0.157	-0.111	-0.106	-0.120	-0.192
-0.235	-0.107	-0.257	-0.201	-0.180	-0.151	-0.164	-0.131
-0.070	-0.262	-0.200	-0.204	-0.175	-0.150	-0.138	-0.146
-0.127	-0.174	-0.211	-0.126	-0.141	-0.146	-0.168	-0.169
-0.010	-0.152	-0.129	-0.151	-0.135	-0.155	-0.202	-0.158
-0.170	-0.168	-0.157	-0.143	-0.146	-0.159	-0.196	-0.149
-0.041	-0.149	-0.157	-0.154	-0.133	-0.096	-0.046	-0.182
-0.071	-0.140	-0.172	-0.163	-0.143	-0.141	-0.080	0.020

Tab. A.3: *Korrelationswerte 444 in Reihenfolge b, a, c, d*

Die Korrelationswerte für die Y Komponente ohne Filterung bei 444:

0.855	-0.231	0.201	-0.154	0.087	-0.118	0.043	0.011
0.431	-0.284	0.213	-0.155	0.165	-0.125	0.105	-0.070
0.326	-0.197	0.195	-0.141	0.157	-0.151	0.091	-0.074
0.232	-0.041	0.099	-0.155	0.092	-0.138	0.042	-0.061
0.340	-0.121	0.094	-0.069	0.135	-0.127	0.100	-0.063
0.433	-0.107	0.083	-0.109	0.103	-0.004	0.014	-0.001
0.455	-0.108	0.016	-0.081	0.092	0.025	0.033	-0.138
0.490	-0.113	0.086	-0.033	0.062	0.012	0.014	-0.141
0.873	0.510	0.423	0.441	0.459	0.556	0.621	0.658
-0.196	-0.179	-0.162	-0.073	-0.041	-0.101	-0.045	-0.064
0.196	0.175	0.073	0.106	0.065	0.078	0.071	0.043
-0.112	-0.080	-0.054	-0.085	-0.092	-0.072	0.003	-0.032
0.118	0.124	0.159	0.075	0.086	0.038	0.093	0.044
-0.097	-0.140	-0.110	-0.117	-0.027	-0.032	0.210	-0.016
0.037	0.086	0.072	0.048	0.026	0.122	0.015	0.084
-0.014	-0.062	-0.043	-0.051	-0.066	-0.124	-0.125	-0.081
0.755	-0.110	0.010	-0.093	-0.042	-0.095	-0.044	-0.011
-0.086	-0.010	-0.138	-0.055	-0.126	-0.077	-0.077	-0.092
-0.014	-0.109	-0.072	-0.132	-0.085	-0.100	-0.059	-0.079
-0.067	-0.065	-0.106	-0.088	-0.129	-0.052	-0.131	-0.081
-0.054	-0.153	-0.057	-0.089	-0.062	-0.081	-0.028	-0.132
-0.107	-0.062	-0.075	-0.060	-0.099	-0.082	-0.057	-0.079
-0.060	-0.103	-0.086	-0.121	-0.076	-0.062	-0.061	-0.018
-0.059	-0.081	-0.100	-0.073	-0.115	-0.158	-0.016	0.022
0.810	-0.078	0.152	-0.006	0.074	0.015	0.036	0.055
-0.041	0.148	0.006	0.075	0.100	0.104	0.075	0.098
0.137	0.009	0.048	0.074	0.091	0.084	0.108	0.089
0.086	0.102	0.070	0.106	0.095	0.129	0.039	0.094
0.095	0.112	0.108	0.129	0.089	0.055	0.040	0.079
0.035	0.115	0.095	0.134	0.107	0.081	-0.006	0.062
0.051	0.083	0.090	0.061	0.099	0.142	0.136	0.061
0.057	0.095	0.084	0.082	0.090	0.064	0.103	0.194

Tab. A.4: *Korrelationswerte 444 in Reihenfolge b, a, c, d*

Die Korrelationswerte für die Cb Komponente ohne Filterung bei 444:

0.849	-0.163	0.149	-0.114	0.052	0.021	0.041	-0.136
0.541	-0.221	0.153	-0.118	0.070	0.023	0.078	-0.062
0.452	-0.146	0.127	-0.075	0.054	-0.079	0.074	-0.030
0.499	-0.077	0.066	-0.039	0.047	-0.044	0.007	-0.031
0.511	-0.062	0.053	-0.040	0.042	-0.074	0.046	-0.029
0.472	-0.093	0.059	-0.045	0.046	0.021	-0.021	-0.027
0.449	-0.105	0.026	-0.027	0.030	0.055	9.80E-5	-0.111
0.547	-0.068	0.046	-0.015	0.022	0.062	-0.013	-0.180
0.844	0.561	0.440	0.385	0.293	0.287	0.360	0.427
-0.209	-0.230	-0.119	-0.063	-0.061	-0.158	-0.108	-0.065
0.156	0.111	0.090	0.041	0.024	0.034	0.026	0.009
-0.066	-0.096	-0.064	-0.033	-0.015	-0.008	0.020	0.014
0.097	0.058	0.048	0.021	0.044	0.024	0.048	0.030
-0.017	-0.015	-0.029	-0.042	-0.020	-0.006	0.181	0.004
0.125	0.072	0.034	0.047	-0.011	0.043	-0.092	0.077
-0.018	-0.025	-0.029	-0.025	-0.039	-0.109	-0.115	-0.113
0.710	-0.081	-0.019	-0.083	-0.045	-0.124	-0.136	-0.162
-0.141	-0.007	-0.105	-0.061	-0.116	-0.047	-0.102	-0.077
0.012	-0.097	-0.049	-0.094	-0.097	-0.086	-0.072	-0.100
-0.063	-0.106	-0.127	-0.100	-0.098	-0.063	-0.148	-0.080
-0.013	-0.106	-0.110	-0.124	-0.067	-0.097	-0.079	-0.111
-0.102	-0.137	-0.107	-0.083	-0.103	-0.097	-0.118	-0.089
-0.087	-0.109	-0.103	-0.098	-0.053	-0.111	-0.082	-0.114
-0.058	-0.075	-0.110	-0.096	-0.088	-0.148	-0.095	0.017
0.767	-0.058	0.102	0.037	0.075	0.059	0.009	-0.044
-0.077	0.127	0.043	0.071	0.060	0.060	0.059	0.101
0.132	0.027	0.095	0.086	0.095	0.100	0.096	0.093
0.035	0.105	0.077	0.133	0.109	0.108	0.056	0.078
0.128	0.097	0.115	0.102	0.112	0.081	0.044	0.080
-0.046	0.060	0.109	0.097	0.095	0.080	-0.016	0.088
0.081	0.083	0.098	0.091	0.112	0.144	0.160	0.031
0.064	0.095	0.088	0.075	0.098	0.063	0.150	0.222

Tab. A.5: *Korrelationswerte 444 in Reihenfolge b, a, c, d*

Die Korrelationswerte für die Cr Komponente ohne Filterung bei 444:

0.858	-0.177	0.114	-0.111	0.070	-0.094	0.021	-0.010
0.509	-0.219	0.165	-0.138	0.083	-0.053	0.042	-0.027
0.475	-0.174	0.132	-0.117	0.080	-0.051	0.043	-0.023
0.513	-0.052	0.098	-0.070	0.049	-0.055	-0.048	-0.042
0.549	-0.099	0.018	-0.021	0.082	0.016	0.079	-0.054
0.534	-0.062	0.087	-0.050	0.056	0.062	-0.039	-0.013
0.544	-0.050	0.013	-0.018	-0.019	0.088	0.007	-0.137
0.565	-0.065	0.074	-0.011	0.034	0.011	-0.025	-0.165
0.855	0.612	0.524	0.498	0.541	0.609	0.573	0.587
-0.192	-0.197	-0.163	-0.060	-0.077	-0.059	-0.035	-0.027
0.150	0.142	0.071	0.056	0.007	0.069	0.034	0.045
-0.066	-0.123	-0.051	-0.047	-0.045	-9.05E-4	0.067	0.014
0.073	0.095	0.061	0.032	0.065	-0.033	0.050	0.060
-0.036	-0.075	-0.019	-0.054	0.031	-0.017	0.274	0.009
0.042	0.060	0.030	0.011	-0.078	0.109	0.019	0.133
0.001	-0.010	-0.020	-0.044	-0.038	-0.146	-0.094	-0.112
0.733	-0.081	-0.020	-0.097	-0.018	-0.116	-0.055	-0.047
-0.056	0.006	-0.093	-0.062	-0.121	-0.080	-0.105	-0.081
0.004	-0.108	-0.061	-0.097	-0.118	-0.084	-0.091	-0.101
-0.045	-0.078	-0.145	-0.093	-0.109	-0.067	-0.180	-0.081
-0.036	-0.124	-0.119	-0.115	-0.066	-0.074	-0.055	-0.160
-0.058	-0.111	-0.083	-0.082	-0.058	-0.123	-0.063	-0.085
-0.042	-0.112	-0.103	-0.099	0.011	-0.021	-0.065	-0.061
-0.045	-0.087	-0.117	-0.081	-0.107	-0.085	-0.069	0.012
0.764	-0.077	0.075	0.014	0.081	-0.029	0.049	0.048
-0.073	0.133	0.045	0.070	0.064	0.107	0.085	0.098
0.117	0.046	0.080	0.067	0.088	0.099	0.085	0.095
0.044	0.109	0.067	0.128	0.097	0.079	0.049	0.030
0.099	0.082	0.094	0.102	0.123	0.002	-0.015	0.069
0.004	0.085	0.097	0.111	0.106	0.101	-0.066	0.081
0.070	0.086	0.085	0.101	0.055	0.155	0.193	0.009
0.059	0.082	0.087	0.062	0.068	0.062	0.119	0.200

Tab. A.6: *Korrelationswerte 444 in Reihenfolge b, a, c, d*

Die Korrelationswerte für die Cb Komponente mit Filterung bei 420:

0.740	-0.222	0.200	-0.047	0.084	-0.061	0.079	-0.030
0.536	-0.170	0.160	-0.063	0.043	-0.056	0.007	-0.058
0.416	-0.162	0.120	-0.051	0.106	-0.080	0.022	-0.013
0.478	-0.054	0.070	-0.042	0.047	-0.038	0.005	-0.018
0.548	-0.072	0.008	-0.025	0.062	-0.005	-0.008	-0.009
0.539	-0.078	0.074	-0.037	0.068	-0.033	-0.003	0.003
0.539	-0.050	-0.023	-0.045	0.037	-0.019	0.004	-0.008
0.571	-0.053	0.027	-0.051	0.001	0.019	-0.003	0.009
0.739	0.520	0.446	0.335	0.280	0.309	0.284	0.320
-0.223	-0.219	-0.133	-0.070	-0.068	-0.096	-0.034	-0.055
0.173	0.101	0.141	0.058	0.029	0.059	-0.015	0.039
-0.107	-0.077	-0.015	-0.089	-0.078	-0.026	0.017	-0.006
0.065	0.092	0.050	0.115	-2.34E-4	-0.011	0.027	-0.034
-0.028	-0.040	-0.046	-0.050	0.022	-0.015	0.040	0.004
0.020	0.064	0.026	0.021	-0.026	0.005	-0.010	-0.011
-0.011	0.011	-0.008	0.018	-0.016	-0.010	-0.017	-0.032
0.547	0.013	0.216	0.028	0.134	0.097	0.147	0.079
-0.006	0.256	0.152	0.258	0.094	0.160	0.129	0.145
0.187	0.195	0.225	0.195	0.197	0.136	0.149	0.148
0.056	0.224	0.233	0.177	0.189	0.175	0.148	0.099
0.096	0.241	0.194	0.158	0.157	0.139	0.129	0.116
0.054	0.156	0.189	0.151	0.131	0.109	0.125	0.129
0.079	0.135	0.127	0.130	0.138	0.132	0.152	0.109
0.089	0.174	0.124	0.137	0.152	0.109	0.101	0.109
0.555	-0.209	-0.040	-0.175	-0.110	-0.149	-0.101	-0.144
-0.207	-0.120	-0.251	-0.188	-0.233	-0.155	-0.133	-0.113
-0.083	-0.244	-0.190	-0.165	-0.146	-0.171	-0.155	-0.162
-0.168	-0.176	-0.218	-0.222	-0.164	-0.160	-0.155	-0.132
-0.045	-0.213	-0.167	-0.178	-0.121	-0.128	-0.106	-0.115
-0.078	-0.178	-0.205	-0.139	-0.132	-0.142	-0.167	-0.127
-0.071	-0.190	-0.131	-0.135	-0.155	-0.119	-0.114	-0.123
-0.083	-0.123	-0.151	-0.085	-0.158	-0.118	-0.107	-0.134

Tab. A.7: *Korrelationswerte 420 in Reihenfolge b, a, c, d*

Die Korrelationswerte für die Cr Komponente mit Filterung bei 420:

0.748	-0.296	0.187	-0.059	0.018	0.056	0.069	-0.018
0.487	-0.190	0.160	-0.109	0.070	-0.043	0.026	-0.028
0.485	-0.124	0.137	-0.075	0.086	-0.060	0.044	-0.058
0.466	-0.067	0.072	-0.065	0.049	-0.037	-0.023	-0.039
0.506	-0.053	-0.014	-0.116	0.033	-0.051	0.013	-0.064
0.539	-0.064	0.075	-0.020	0.027	-0.070	0.003	-0.033
0.543	-0.043	0.041	-0.005	0.017	-0.008	0.005	-0.016
0.562	-0.050	-0.025	-0.048	0.016	0.004	0.022	0.020
0.747	0.522	0.516	0.501	0.477	0.571	0.550	0.562
-0.221	-0.206	-0.136	-0.088	-0.062	-0.096	-0.093	-0.002
0.179	0.169	0.132	0.080	-0.057	0.085	0.001	0.070
-0.057	-0.137	-0.025	-0.072	-0.021	-0.056	0.011	0.025
0.053	0.107	0.084	0.172	0.045	0.013	0.029	0.042
-0.007	-0.008	-0.014	-0.057	-0.066	-0.044	-0.015	-0.032
0.015	0.032	0.002	-0.032	3.24E-4	0.008	0.019	0.013
-0.007	-0.009	0.027	0.002	0.001	0.044	-0.016	8.35E-4
0.568	-0.027	0.209	0.136	0.067	0.147	0.120	0.064
0.056	0.252	0.245	0.260	0.153	0.172	0.155	0.114
0.240	0.184	0.315	0.236	0.176	0.154	0.182	0.108
0.117	0.249	0.176	0.139	0.190	0.210	0.124	0.112
0.110	0.195	0.154	0.118	0.150	0.161	0.156	0.141
0.087	0.120	0.193	0.125	0.102	0.137	0.129	0.139
0.074	0.181	0.158	0.146	0.125	0.147	0.159	0.106
0.103	0.186	0.135	0.148	0.179	0.135	0.130	0.127
0.540	-0.272	-0.065	-0.131	-0.102	-0.041	-0.050	-0.090
-0.230	-0.152	-0.284	-0.224	-0.214	-0.153	-0.148	-0.123
-0.072	-0.260	-0.264	-0.259	-0.206	-0.197	-0.226	-0.180
-0.130	-0.188	-0.252	-0.213	-0.197	-0.185	-0.132	-0.161
-0.065	-0.171	-0.186	-0.229	-0.110	-0.134	-0.147	-0.135
-0.086	-0.163	-0.180	-0.110	-0.105	-0.104	-0.179	-0.133
-0.095	-0.185	-0.134	-0.142	-0.110	-0.137	-0.136	-0.132
-0.088	-0.142	-0.152	-0.133	-0.125	-0.106	-0.152	-0.122

Tab. A.8: *Korrelationswerte 420 in Reihenfolge b, a, c, d*

Die Korrelationswerte für die Cb Komponente ohne Filterung bei 420:

0.738	-0.209	0.180	-0.033	0.071	-0.056	0.077	-0.029
0.552	-0.161	0.128	-0.064	0.053	-0.053	0.004	-0.059
0.442	-0.154	0.102	-0.039	0.099	-0.069	0.014	-0.015
0.501	-0.046	0.069	-0.039	0.039	-0.033	9.35E-4	-0.009
0.558	-0.073	-0.007	-0.020	0.059	-0.004	-7.35E-4	-0.022
0.549	-0.083	0.073	-0.038	0.060	-0.038	-0.003	0.009
0.550	-0.047	-0.030	-0.047	0.035	-0.023	0.005	-0.008
0.578	-0.057	-0.004	-0.047	-0.005	0.025	-9.57E-4	0.017
0.738	0.540	0.465	0.354	0.293	0.323	0.295	0.335
-0.204	-0.207	-0.126	-0.056	-0.084	-0.100	-0.024	-0.049
0.162	0.081	0.123	0.047	0.023	0.050	-0.008	0.045
-0.097	-0.062	-0.013	-0.089	-0.076	-0.012	0.016	0.002
0.066	0.082	0.051	0.117	-0.009	-0.013	0.034	-0.032
-0.023	-0.033	-0.036	-0.042	0.016	-0.003	0.042	0.001
0.020	0.052	0.029	0.021	-0.027	-0.005	-0.022	-0.007
-0.010	0.008	-0.019	0.031	-0.015	-0.004	-0.016	-0.027
0.504	-0.142	0.015	-0.149	-0.060	-0.081	-0.044	-0.105
-0.150	0.018	-0.170	-0.017	-0.140	-0.088	-0.116	-0.087
-0.007	-0.103	-0.072	-0.097	-0.069	-0.098	-0.091	-0.099
-0.111	-0.044	-0.063	-0.095	-0.069	-0.086	-0.090	-0.159
-0.052	-0.027	-0.069	-0.092	-0.077	-0.102	-0.108	-0.115
-0.085	-0.101	-0.075	-0.096	-0.128	-0.125	-0.136	-0.115
-0.057	-0.120	-0.095	-0.117	-0.115	-0.093	-0.094	-0.128
-0.051	-0.063	-0.129	-0.089	-0.096	-0.110	-0.134	-0.112
0.596	-0.058	0.151	-0.017	0.086	0.022	0.094	0.039
-0.056	0.130	0.059	0.101	-0.016	0.091	0.109	0.119
0.119	0.034	0.102	0.134	0.118	0.064	0.090	0.091
-0.001	0.101	0.067	0.048	0.090	0.105	0.078	0.119
0.100	0.050	0.101	0.074	0.122	0.111	0.134	0.113
0.060	0.068	0.061	0.100	0.118	0.098	0.085	0.119
0.064	0.059	0.101	0.105	0.091	0.102	0.126	0.105
0.055	0.114	0.090	0.140	0.091	0.107	0.129	0.087

Tab. A.9: *Korrelationswerte 420 in Reihenfolge b, a, c, d*

Die Korrelationswerte für die Cr Komponente ohne Filterung bei 420:

0.746	-0.285	0.164	-0.042	0.011	0.054	0.063	-0.011
0.505	-0.192	0.137	-0.090	0.071	-0.038	0.036	-0.026
0.499	-0.114	0.108	-0.079	0.073	-0.056	0.041	-0.062
0.486	-0.069	0.054	-0.065	0.048	-0.024	-0.022	-0.038
0.522	-0.046	-0.025	-0.117	0.027	-0.043	0.004	-0.077
0.549	-0.062	0.071	8.07E-4	0.038	-0.063	0.007	-0.023
0.552	-0.047	0.035	-0.002	0.010	-0.001	0.003	-0.018
0.563	-0.052	-0.063	-0.055	0.013	0.009	0.020	0.027
0.747	0.547	0.544	0.532	0.502	0.580	0.553	0.564
-0.202	-0.205	-0.122	-0.094	-0.055	-0.096	-0.095	0.001
0.153	0.153	0.105	0.071	-0.059	0.073	0.005	0.089
-0.044	-0.125	-0.014	-0.060	-0.019	-0.043	0.006	0.040
0.046	0.088	0.091	0.164	0.024	-0.001	0.036	0.055
-0.003	-0.003	-0.014	-0.066	-0.073	-0.037	-0.014	-0.024
0.012	0.024	0.007	-0.033	1.69E-4	0.006	0.019	0.017
-0.012	-0.017	0.016	0.005	-0.005	0.045	-0.022	0.007
0.530	-0.183	0.014	-0.032	-0.089	0.020	-0.014	-0.059
-0.102	0.007	-0.079	-0.020	-0.115	-0.051	-0.082	-0.105
0.015	-0.145	-0.055	-0.122	-0.084	-0.093	-0.073	-0.151
-0.059	-0.034	-0.155	-0.136	-0.068	-0.045	-0.105	-0.139
-0.045	-0.071	-0.115	-0.130	-0.086	-0.081	-0.105	-0.115
-0.040	-0.120	-0.087	-0.119	-0.147	-0.096	-0.111	-0.110
-0.061	-0.068	-0.097	-0.094	-0.122	-0.088	-0.098	-0.131
-0.027	-0.055	-0.122	-0.089	-0.061	-0.113	-0.103	-0.098
0.578	-0.129	0.107	0.039	0.053	0.083	0.082	0.033
-0.075	0.106	0.048	0.061	0.061	0.070	0.079	0.096
0.136	0.042	0.097	0.071	0.060	0.047	0.035	0.056
0.054	0.112	0.054	0.042	0.050	0.074	0.094	0.091
0.092	0.095	0.077	0.015	0.128	0.110	0.112	0.121
0.054	0.077	0.098	0.134	0.136	0.136	0.059	0.115
0.041	0.066	0.133	0.097	0.147	0.101	0.121	0.107
0.042	0.092	0.103	0.098	0.121	0.130	0.088	0.104

Tab. A.10: *Korrelationswerte 420 in Reihenfolge b, a, c, d*

A.3 Graphiken und Tabellen

Vergleiche hierzu auch die Bilder aus Kapitel 2.2.2, S. 9 u. S.10

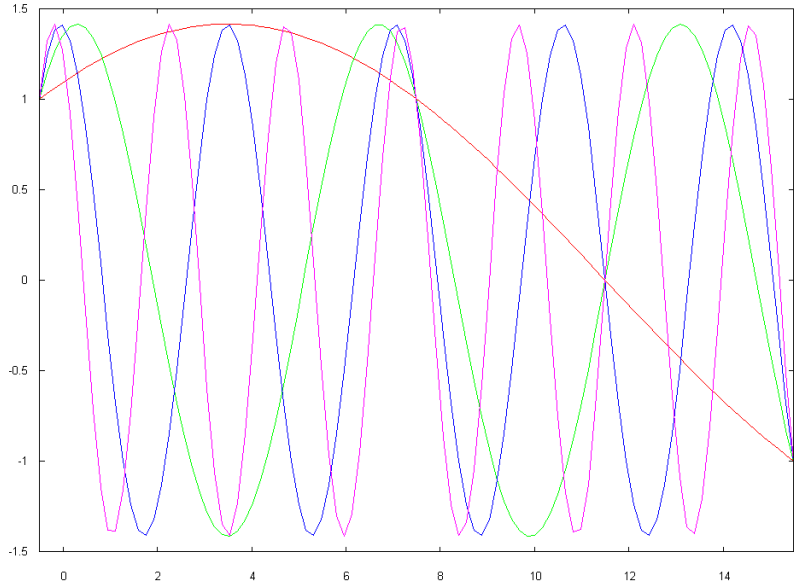


Abb. A.5: *Eindimensionale Basisfunktionen der LHT ohne Modulationsfenster für gerade, negative Frequenzen k = 0, -2, -4, -6*

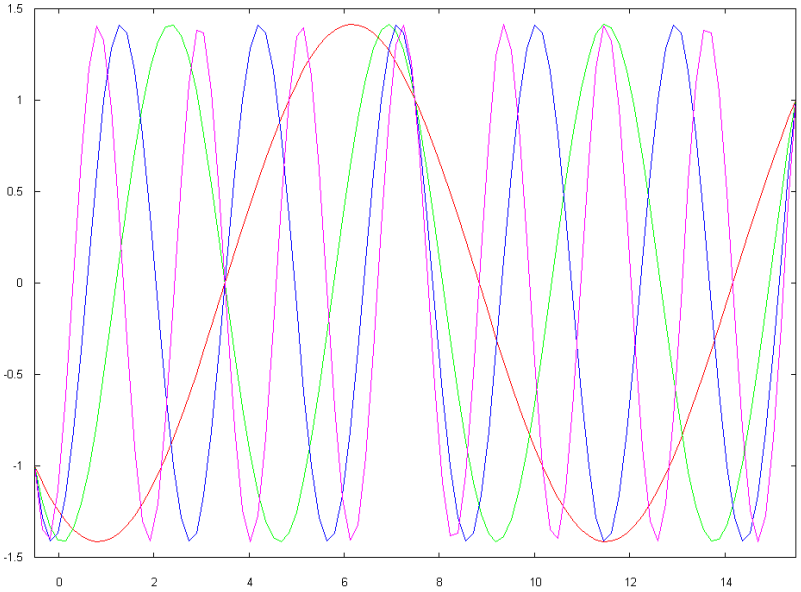


Abb. A.6: *Eindimensionale Basisfunktionen der LHT ohne Modulationsfenster für unerade, negative Frequenzen k = -1, -3, -5, -7*

Die folgenden 12 Graphen und Tabellen gehören zum Bild *Peppers*. *PSNR über Dateigröße* ist in halblogarithmischer Darstellung. Die Angabe ImageJ in der Legende bedeutet, daß der *ImageJ-Jpeg-Dekoder* verwendet wurde.

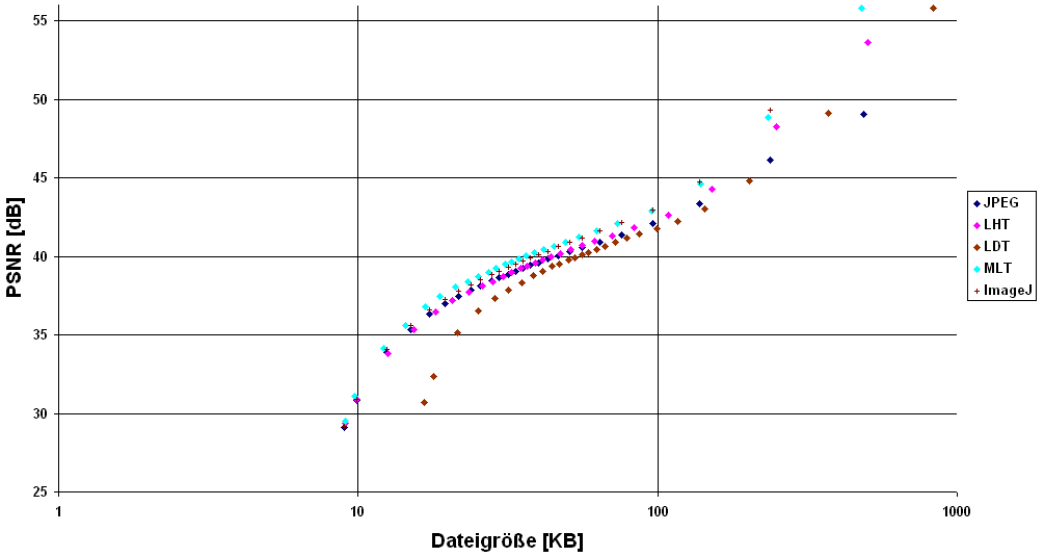


Abb. A.7: *PSNR über Dateigröße; Peppers 4:4:4*

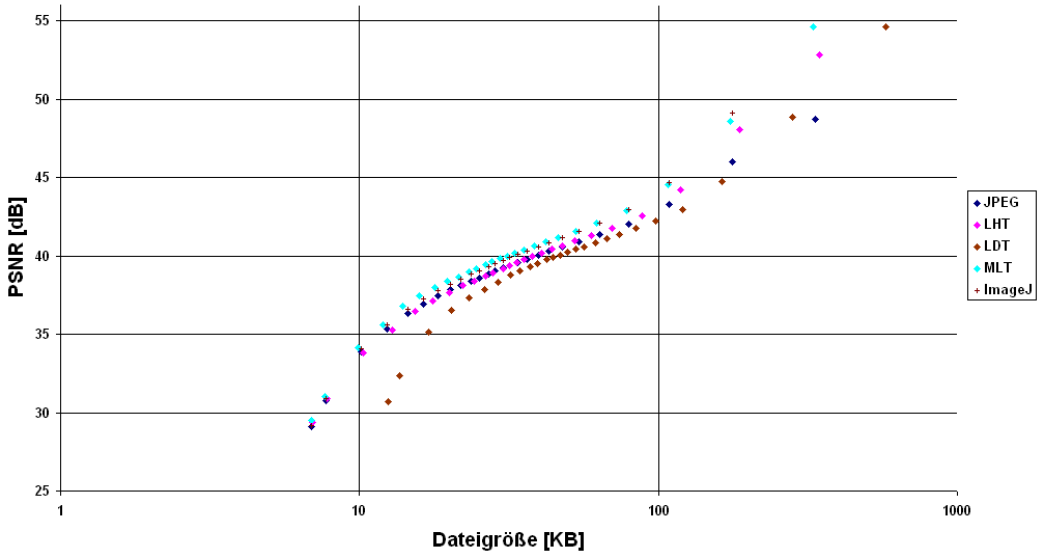


Abb. A.8: *PSNR über Dateigröße; Peppers 4:2:2*

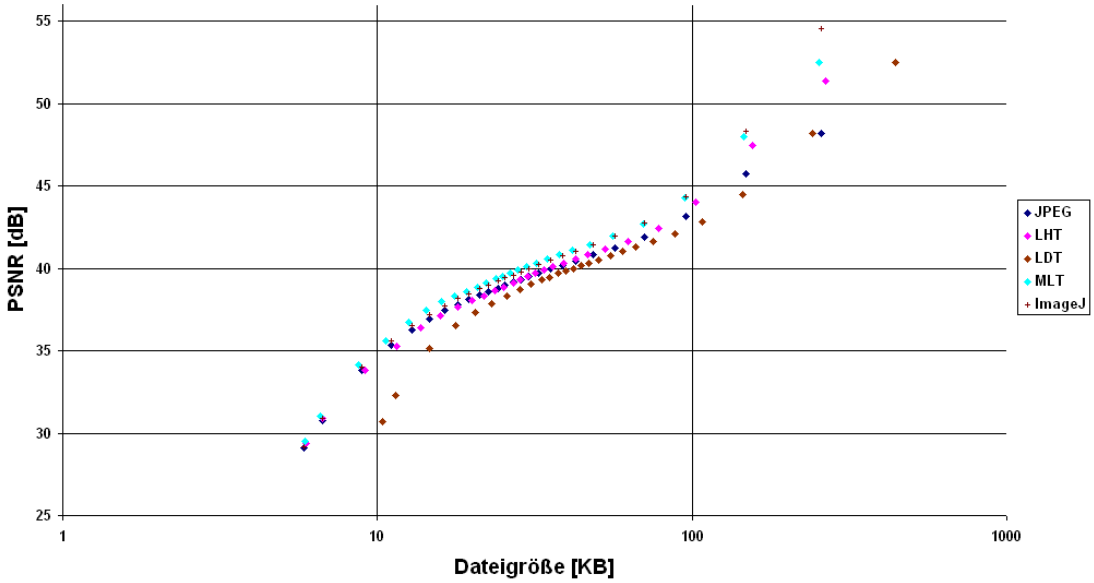


Abb. A.9: *PSNR über Dateigröße; Peppers 4:1:1*

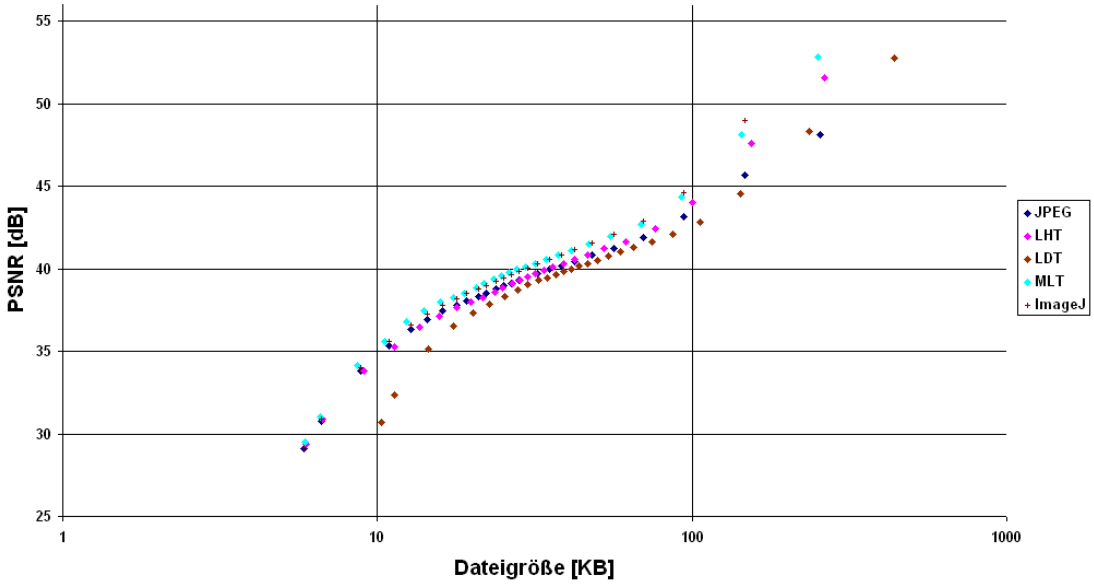


Abb. A.10: *PSNR über Dateigröße; Peppers 4:2:0*

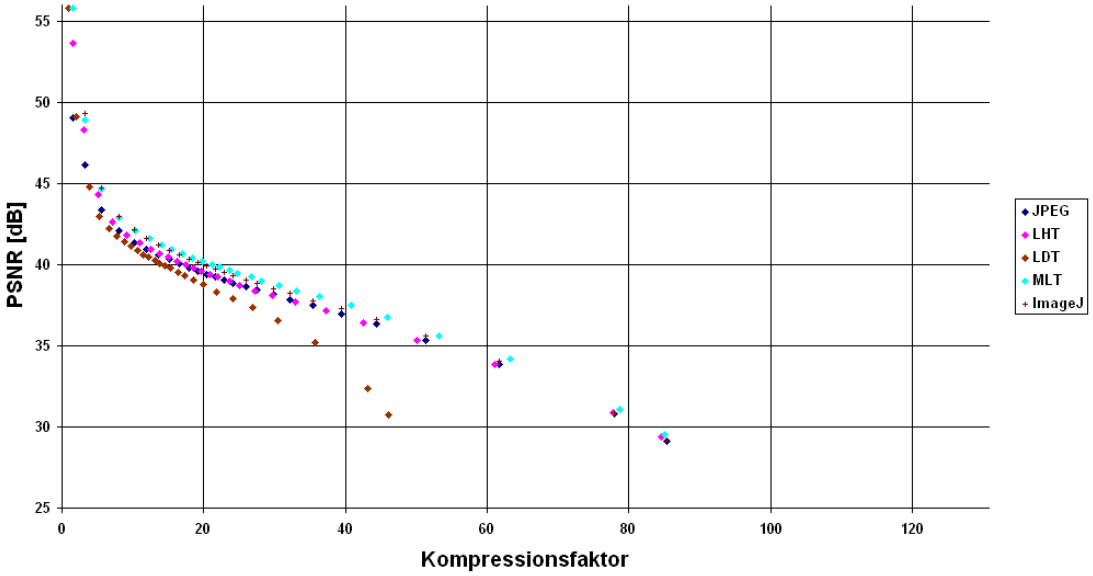


Abb. A.11: *PSNR über KF; Peppers 4:4:4*

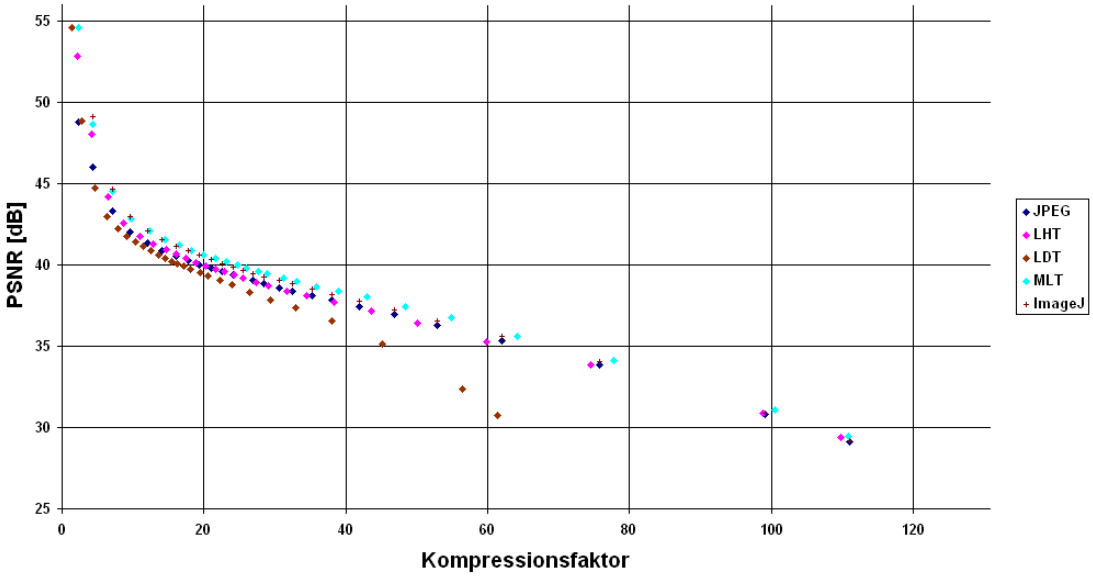


Abb. A.12: *PSNR über KF; Peppers 4:2:2*

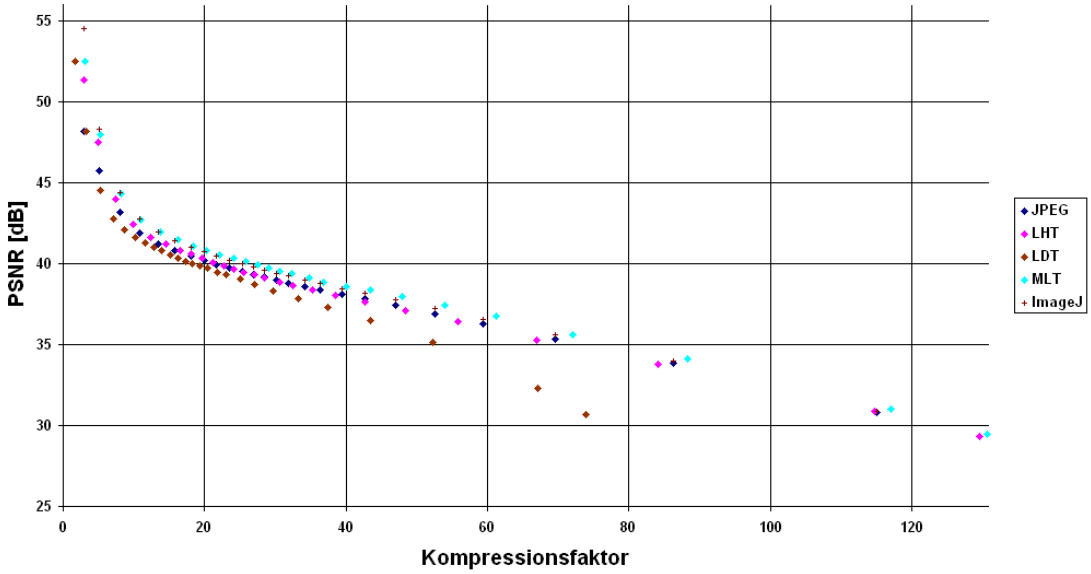


Abb. A.13: *PSNR über KF; Peppers 4:1:1*

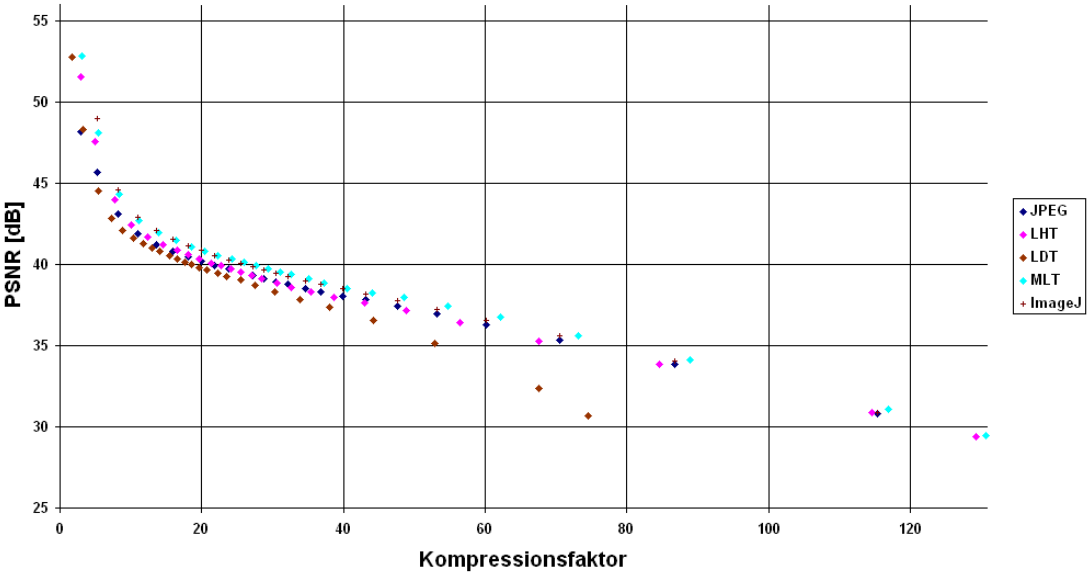


Abb. A.14: *PSNR über KF; Peppers 4:2:0*

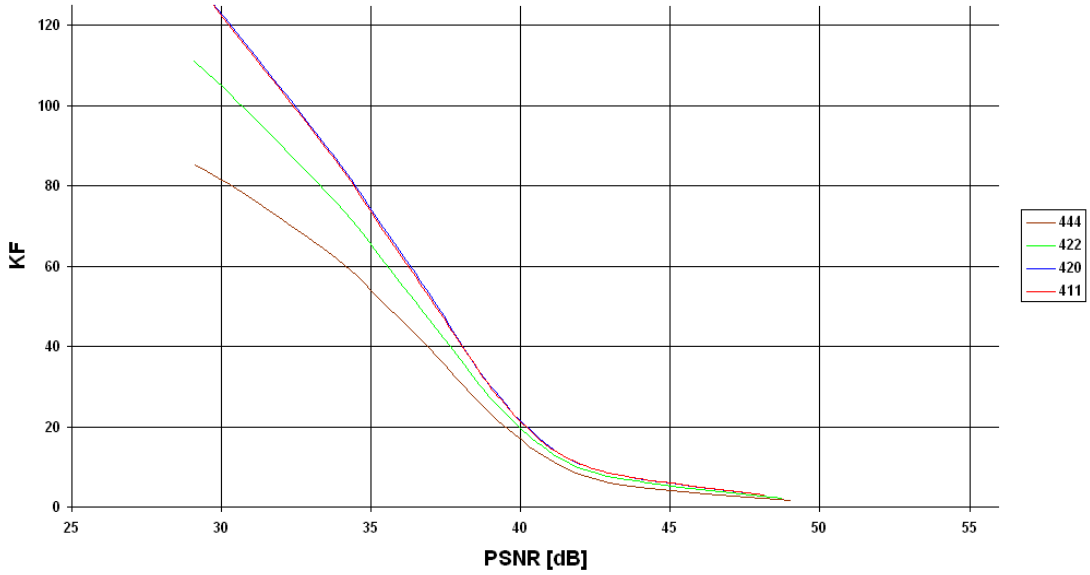


Abb. A.15: *KF über PSNR; Peppers; JPEG*

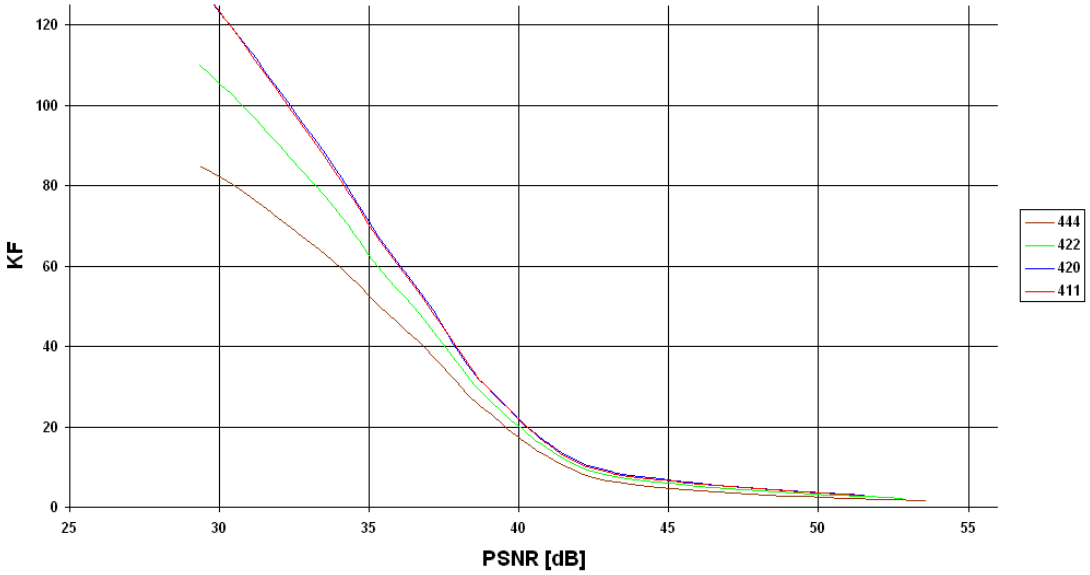


Abb. A.16: *KF über PSNR; Peppers; LHT*

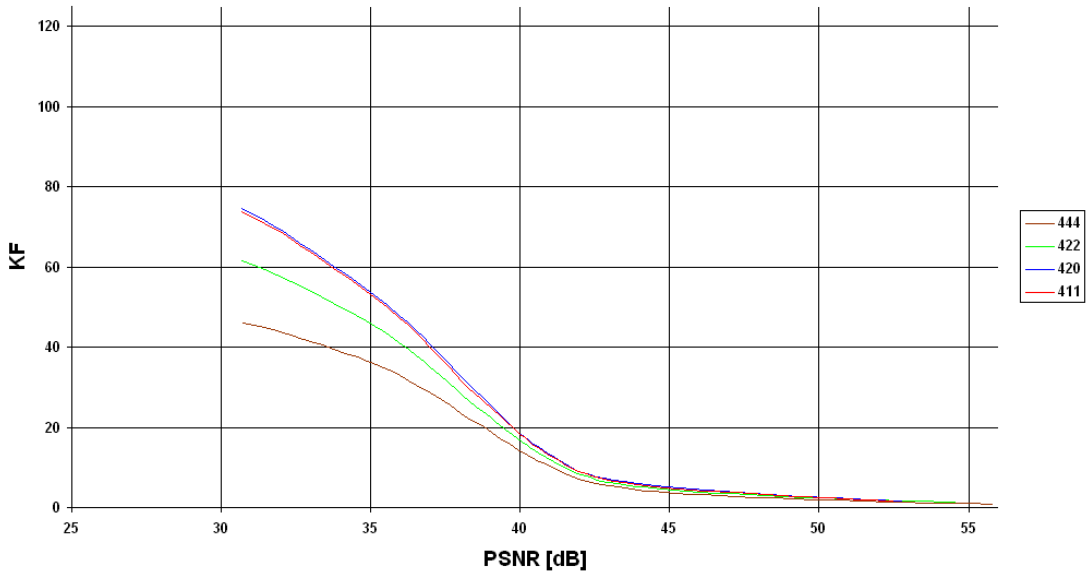


Abb. A.17: *KF über PSNR; Peppers; LDT*

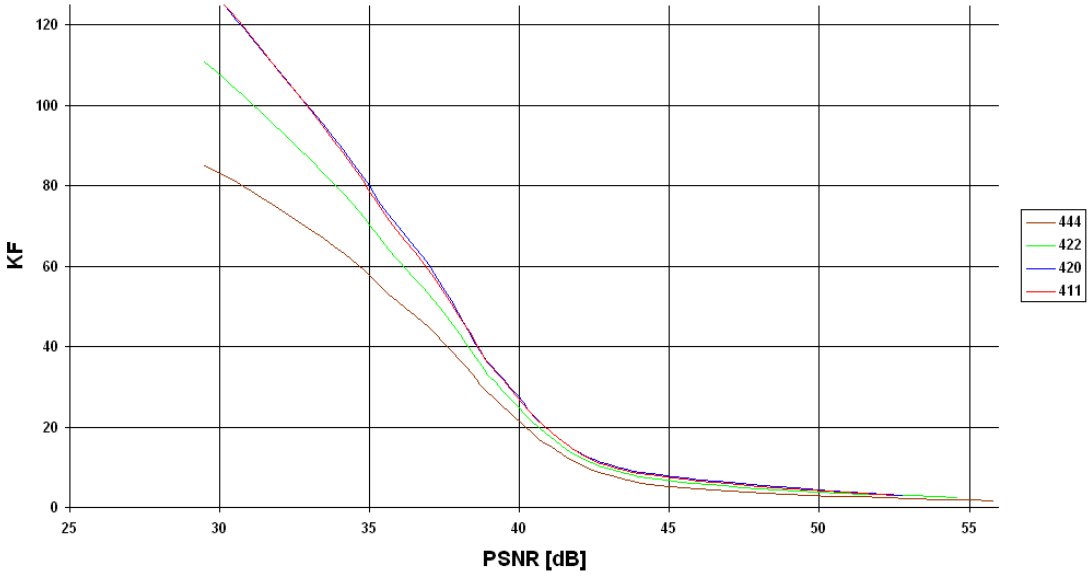


Abb. A.18: *KF über PSNR; Peppers; MLT*

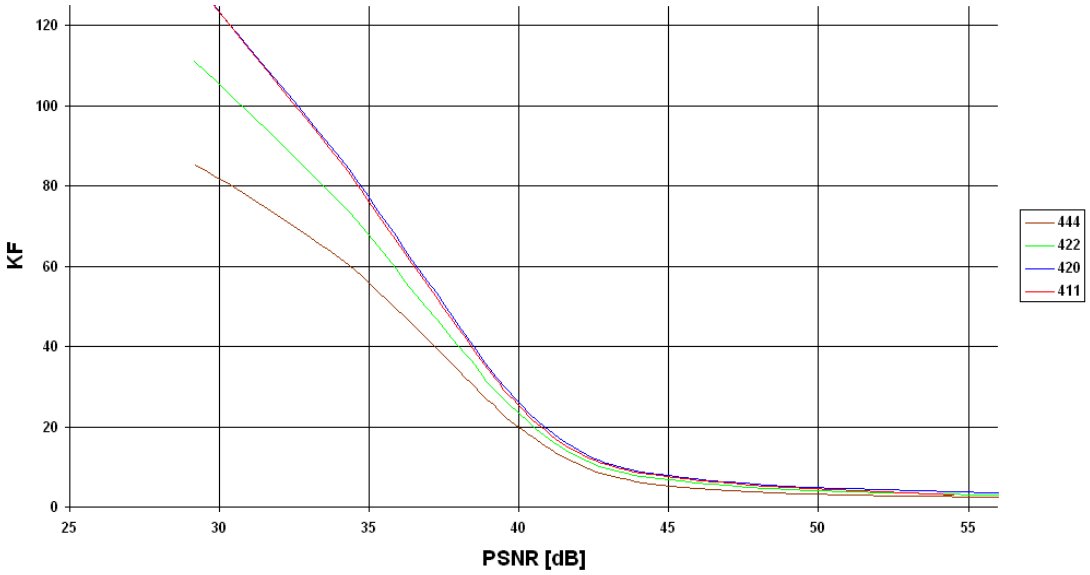


Abb. A.19: *KF über PSNR; Peppers; JPEG mit dem ImageJ Dekoder*

Die folgende Tabelle zeigt die abweichenden JPEG PSNR Werte des *ImageJ-Dekoders* zum *JPEGDecoder* vom Bild Peppers.

JD = JpegDecoder; IJ = ImageJ Dekoder; q = Qualitätsfaktor

	4:4:4		4:2:2		4:1:1		4:2:0	
q	JD	IJ	JD	IJ	JD	IJ	JD	IJ
0	29,11	29,18	29,10	29,17	29,10	29,17	29,09	29,17
4	30,80	30,89	30,79	30,87	30,78	30,86	30,79	30,87
8	33,85	34,05	33,85	34,04	33,82	34,01	33,84	34,04
12	35,34	35,61	35,33	35,60	35,31	35,58	35,32	35,60
16	36,32	36,59	36,31	36,58	36,29	36,55	36,30	36,57
20	36,96	37,26	36,94	37,25	36,91	37,21	36,93	37,25
24	37,47	37,79	37,45	37,78	37,42	37,74	37,44	37,77
28	37,85	38,21	37,83	38,20	37,80	38,16	37,80	38,19
32	38,14	38,53	38,13	38,52	38,09	38,47	38,07	38,50
36	38,42	38,83	38,41	38,82	38,37	38,76	38,34	38,79
40	38,62	39,03	38,60	39,02	38,56	38,96	38,52	38,99
44	38,86	39,30	38,84	39,28	38,80	39,22	38,77	39,26
48	39,04	39,49	39,02	39,47	38,97	39,41	38,95	39,46
52	39,23	39,70	39,21	39,68	39,16	39,60	39,13	39,66
56	39,41	39,89	39,39	39,88	39,33	39,79	39,32	39,86
60	39,60	40,10	39,58	40,09	39,51	40,00	39,51	40,07
64	39,81	40,32	39,78	40,31	39,71	40,21	39,71	40,29
68	40,05	40,60	40,03	40,58	39,96	40,48	39,95	40,56
72	40,30	40,88	40,27	40,86	40,19	40,75	40,19	40,84
76	40,58	41,18	40,55	41,16	40,46	41,03	40,46	41,14
80	40,93	41,60	40,90	41,57	40,81	41,44	40,80	41,55
84	41,37	42,14	41,34	42,12	41,24	41,96	41,23	42,09
88	42,07	42,98	42,03	42,94	41,91	42,75	41,90	42,91
92	43,38	44,72	43,31	44,66	43,14	44,36	43,13	44,61
96	46,15	49,32	46,01	49,11	45,70	48,32	45,68	48,98
100	49,02	63,62	48,74	59,68	48,17	54,53	48,15	58,22

Tab. A.11: *PSNR Werte Peppers JPEG*

Die unterschiedliche Auswirkung der beiden JPEG Dekoder am Beispiel der *KF über PSNR* Diagramme von *Blocks*.

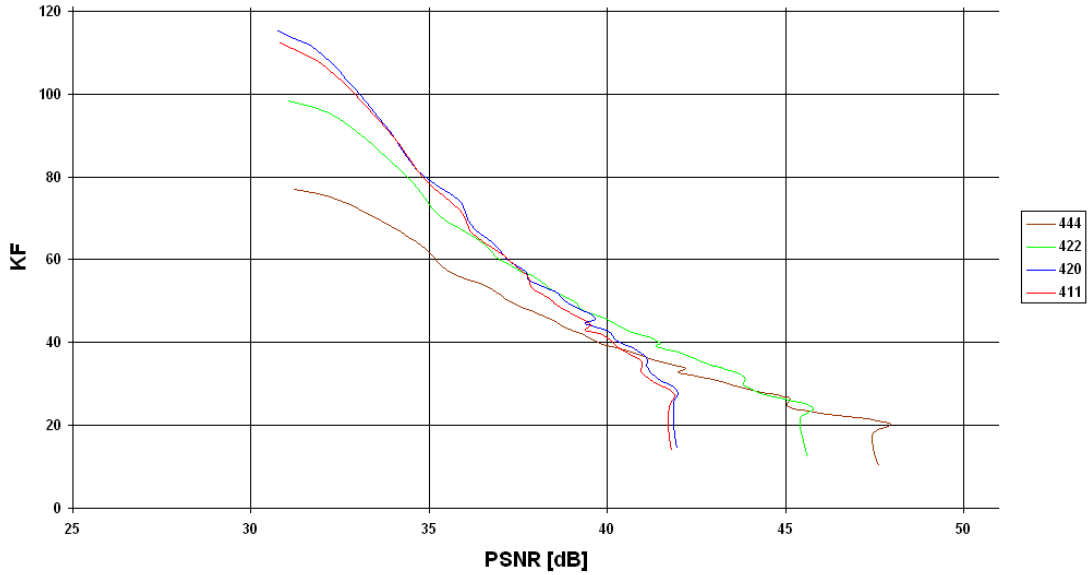


Abb. A.20: *KF über PSNR; Blocks; JPEG mit dem JpegDecoder*

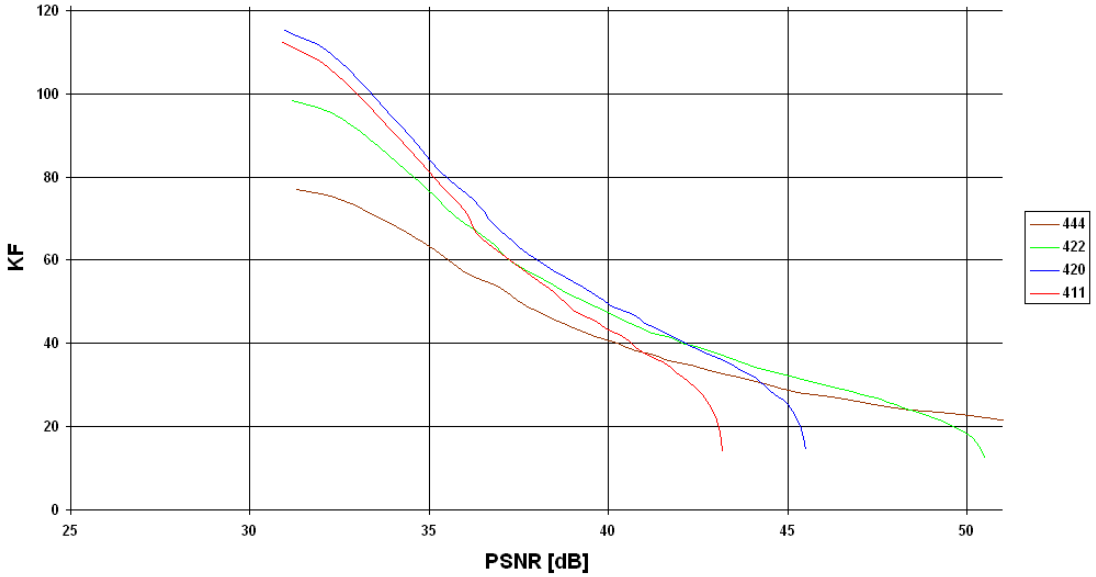


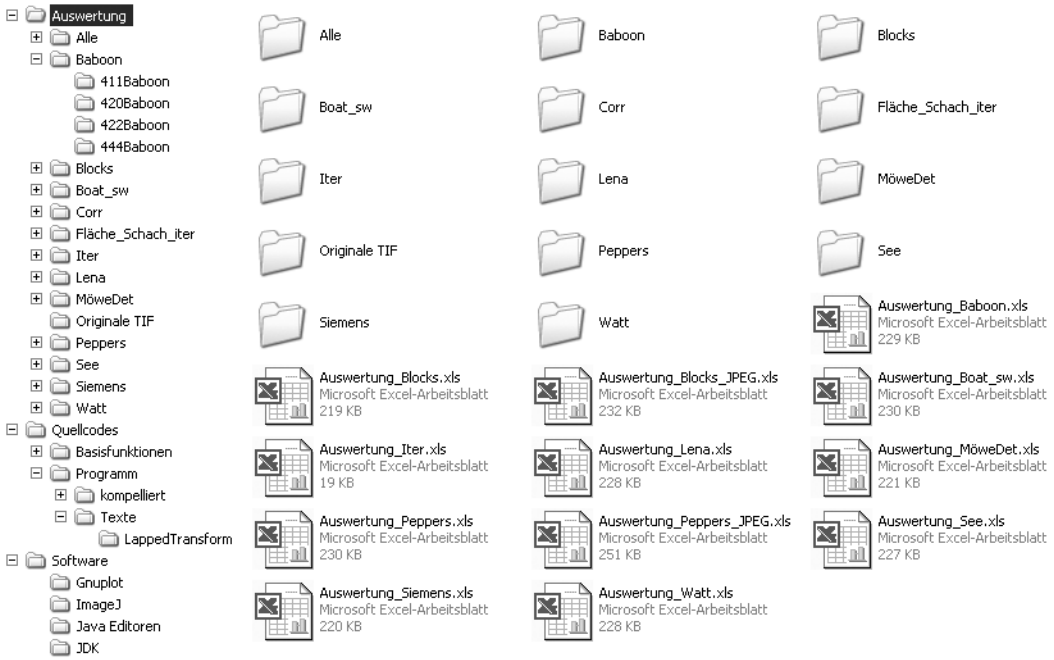
Abb. A.21: *KF über PSNR; Blocks; JPEG mit dem ImageJ Dekoder*

A.4 Inhalt der Compact Disc

Die beiligende CD enthält folgende Daten:

- Diplomarbeit *Diplom LHT* im PDF-Format
- Alle Quelltexte
- Alle kodierten Bilder und die dazugehörigen Excel Tabellen und Graphen
- Die verwendete Software

Die Bilder und Tabellen befinden sich im Ordner *Auswertungen* in den jeweiligen Unterordnern.
Die Messungen der Korrelationswerte werden im Ordner *Corr* gefunden.
Siehe dazu folgendes Ordnerstrukturgramm.



Danksagung

Ich danke Prof. Dr. Dietmar Kunz und Prof. Dr. Gregor Fischer für die Übernahme der Korrektur dieser Arbeit.

Vor allem möchte ich mich bei Prof. Kunz für die sehr gute Betreuung von fachlicher und kollegialer Seite bedanken.

Mein besonderer Dank gilt meinen Eltern für die jahrelange Unterstützung während meiner Ausbildungszeit.

Außerdem danke ich Ulrich Vollenbruch, Sylvia Schmitt und Stephanie Meyen für die Durchsicht dieser Arbeit.

Eidesstattliche Erklärung

Ich versichere hiermit, die vorgelegte Arbeit in dem gemeldeten Zeitraum selbstständig verfasst und mich keiner anderen als der angegebenen Hilfsmittel und Quellen bedient zu haben.

Köln, den 04.07.2006

(Stephan May)

Sperrvermerk

Die vorgelegte Arbeit unterliegt keinem Sperrvermerk.

Weitergabeerklärung

Ich erkläre hiermit mein Einverständnis, daß das vorliegende Exemplar meiner Diplomarbeit oder eine Kopie hiervon für wissenschaftliche Zwecke verwendet werden darf.

Köln, den 04.07.2006

(Stephan May)